



Swiss Federal Institute of Technology Zurich

Seminar for
Statistics

Department of Mathematics

Master Thesis

Fall 2018

Christopher Salahub

Seen to be done

A statistical investigation of peremptory challenge

Submission Date: March 3 2019

Co-Adviser:

Adviser: Prof. Dr. Marloes Maathuis

Preface

This work would be nowhere near as polished or complete without the effort of Prof. Dr. Marloes Maathuis to ensure I was performing analysis with a clear direction and purpose. I would like to thank her for finding time in her busy schedule to allow for weekly meetings. The group meetings organized by her Ph.D. student Marco Eigenmann were also critical in the development of more nuanced analysis and intuitive visualizations. I thank Marco Eigenmann for organizing them, and Jinzhou Li, Armin Fingerle, Sanzio Monti, and Qikun Xiang for attending my presentations and listening attentively. A special thanks is extended to Cédric Bleutler and Leonard Henckel, both of whom were especially engaged and participated in lengthy discussions both during and outside of the group meetings.

I would like to acknowledge in particular Prof. Dr. Tilman Altwicker for his detailed suggestions on where to look for more legal context on the topic and Prof. Dr. Samuel Baumgartner for his research suggestions. They were very important at providing the necessary information to begin a first investigation of the topic. Of course, without the cooperation of Dr. Ronald Wright, Dr. George Woodworth, Dr. Barbara O'Brien, and Dr. Catherine Grosso for generously providing me with the data from their investigations on the subject of preemptory challenge. Without this data, the visualizations and modelling presented here simply would not have been possible, and so I am exceptionally grateful that they were so enthusiastic to share the fruits of their labour to help cultivate mine.

Abstract

Short summary of my thesis.

Contents

Notation	xi
1 Introduction	1
2 Peremptory Challenges	3
2.1 Jury Selection Procedures	3
2.2 The Role of the Jury	5
2.3 Modern Peremptory Challenge Controversy	5
2.4 The Role of the Peremptory Challenge	7
2.5 History	7
2.5.1 Pre-English History	8
2.5.2 In English Law (1066–1988)	8
2.5.3 In American Law (ca. 1700–1986)	9
2.5.4 In Canadian Law (ca 1800–2018)	10
2.6 Summary	11
3 Data	13
3.1 Jury Sunshine Project	13
3.1.1 Methodology	13
3.1.2 Cleaning	14
3.1.3 Variable Synthesis	18
3.2 Stubborn Legacy Data	18
3.2.1 Methodology	18
3.2.2 Cleaning	19
3.3 Philadelphia Data	19
3.3.1 Methodology	19
3.3.2 Cleaning	19
4 Empirical Analysis	21
4.1 Extremes of Partiality	21
4.2 The Impact of Race	22
4.3 Other Factors	26
4.4 Modelling	29
4.5 Prosecution and Judge Homogeneity	31
4.6 Case Level Summary	31
4.7 Useful Quoted Bits	32
5 Summary	33
5.1 Future Work	33
Bibliography	35
A Developing an Effective Visualization of Conditional Probability	39
A.1 The Mobile Plot	41
B Complementary information	43
B.1 Jury Sunshine Irregularities	43

B.2	Jury Sunshine Charge Classification	45
B.3	Using Sweave to include R code (and more) in your report	45
C	Mathematical Results	47
C.1	Conditional Distribution of a Poisson Expectation Given Marginal Counts .	47
D	Code	49
D.1	Data Processing Code	49
D.2	Analysis Code	64
	Epilogue	83

List of Figures

3.1	Simple Charge Tree Example	17
4.1	The “Mobile Plot” of Racial Combination and Strikes	23
4.2	Racial Combination and Strikes with Confidence Intervals	25
4.3	Political Affiliation by Race and Gender (Sunshine)	27
4.4	Strikes by Political Affiliation, Race, and Defendant Race (Sunshine)	28
4.5	Strike Source by Race and Gender (Sunshine)	29
4.6	Strike Source by Race and Gender (Sunshine)	30
4.7	Lawyer Experience (Sunshine)	32
A.1	Mosaic Plot of Defendant and Venire Member Race	40
A.2	First Parallel Coordinate Attempt	41
B.1	Regular expression charge tree visualized	44

List of Tables

4.1	Implied Rejection Boundaries	22
4.2	Strike Rate by Race	23
B.1	Jury Sunshine Irregularities	43

Notation and Terms

In order to facilitate clarity despite brevity, a list of terms used in this paper is presented here.

Prosecution/State The legal representation which argues for conviction

Defence The legal representation which argues against conviction

Court Reference to the judge, prosecution, and defence

Venire The population sample from which a jury is selected (according to [Mirriam-Webster \(2019a\)](#) derived from the latin *venire facias*: “may you cause to come”)

Jury The final group of (usually) twelve chosen venire members which judge the guilt or innocence of the accused/defendant

Accused/Defendant The individual on trial for a crime

Voir dire From old French “to speak the truth” (see [Mirriam-Webster \(2019b\)](#)), this is the questioning process used by the court to assess the suitability of a venire member to sit on the jury

Struck In the context of a venire member being rejected from the jury, struck indicates removal by peremptory challenge or challenge with cause

Litigants The accuser and the accused

Chapter 1

Introduction

The Gerald Stanley murder trial was noteworthy for all of the wrong reasons. The first reason was the crime itself. The rural region around Biggar, Saskatchewan ([Quenneville \(2018\)](#)) is not known for crime, indeed, the crime statistics collected by Statistics Canada suggest it is one of the safest in the province ([Statistics Canada \(2018\)](#)). Any murder at all would be worthy of attention and subject to plenty of drama. But beyond the damage this trial has done to the community, this trial is noteworthy because it led to a significant re-examination of the legal jurisprudence surrounding the jury selection process culminating in the proposition of Bill C-75 by the Canadian government in March of 2018 ([42nd Parliament of Canada \(2018a\)](#)), less than two months after the trial’s verdict ([Quenneville and Warick \(2018\)](#)).

Bill C-75, in part, aims to ameliorate one of the critical points of contention about the Gerald Stanley case: the use of peremptory challenges in jury selection. The outsized impact of the case was due, in large part, to its racial aspect. Gerald Stanley, a white man, was accused of second degree murder in the killing of Colten Boushie, a First Nations man. Given Canada’s troubled history with First Nations groups, this alone would have been enough to make the trial a flash point for race issues, but that was not the worst aspect of the trial. Rather, it was the alleged use of peremptory challenges to strike five potential jurors who “appeared” to be First Nations, resulting in an all-white jury, that proved to be the most controversial and influential facet of the entire affair ([Harris \(2018\)](#), [MacLean \(2018\)](#)).

With Bill C-75 currently moving through the Canadian parliamentary system, having completed its second reading in June 2018 ([42nd Parliament of Canada \(2018b\)](#)), a close re-examination of the practice of peremptory challenge is warranted. A great deal of ink has already been spilled on both sides of the debate (see [Hasan \(2018\)](#), [Zinchuk \(2018\)](#), and [Roach \(2018\)](#)), but startlingly little of this discussion has been based on any hard evidence on the impact of peremptory challenge in jury selection. This paper aims to provide analysis and evidence to illuminate the topic further by analyzing three separate peremptory challenge data sets collected in the United States, namely [Wright, Chavis, and Parks \(2018\)](#), [Grosso and O’Brien \(2012\)](#), and [Baldus, Woodworth, Zuckerman, and Weiner \(2001\)](#). While this data cannot tell us if challenges were racially motivated in the Stanley trial, stepping back from this fraught legal episode to take a wider view of the practice of peremptory challenge provides a more sober place to start the discussion of its place in modern jury trials.

This paper will proceed in five parts. Chapter 2 provides a brief history of the practice of peremptory challenges in jury trials, in particular explaining their original motivation, past implementations, and how they have developed in the United States, the United Kingdom, and Canada. Chapter 3 proceeds to discuss the three data sets obtained, explaining the sources and collection methods before detailing the cleaning and preprocessing. Chapter 4 then provides the details and results of the analysis performed on the different data sets. It begins discussing the Jury Sunshine data set, which was used as a 'test' set of sorts, where analysis could be flexibly performed before the final analysis methods were turned to the other two data sets. The results of this analysis are compared to previous works in Chapter ???. Finally, the results and findings are summarized in ??, and recommendations based on the observations obtained here are provided.

Chapter 2

Peremptory Challenges

The focus of this text is the practice of peremptory challenges in a jury trial system, a highly specific practice in a particular context which may not be known in detail to the reader. As a consequence, a brief exploration of their history, motivation, and current use is presented here. It is not meant to be exhaustive, but rather to provide context and references for an interested and motivated reader to learn more. Indeed, many details have been omitted from the summary of the history in particular.

2.1 Jury Selection Procedures

Before reviewing the history, it is best to give some context and an explanation for readers unfamiliar with the jury system and general courtroom procedures. While the process of jury selection varies by jurisdiction and crime severity, the general steps shared by jury trials are outlined below. More detail and a discussion of the diversity of jury selection procedures can be found in [Ford \(2010\)](#), [Hans and Vidmar \(1986\)](#), and [Van Dyke \(1977\)](#). To select a jury:

- i.) Eligible individuals are selected at random from the population (using a list known as the *jury roll*) of the region surrounding the location of the crime, the sampled individuals are called the *venire*
- ii.) The venire is presented to the court, either as a group or sequentially (borrowing the names of [Ford \(2010\)](#): the “struck-jury” system and the “sequential-selection” system, respectively)
- iii.) The presented venire member(s) are questioned in a process called *voir dire*, which can result in three possible outcomes for each venire member:
 - (a) The venire member is removed with cause, the cause provided by either the prosecutor or defence lawyer and admitted by the judge
 - (b) The venire member is removed by a *peremptory challenge* by the prosecutor or defence lawyer, where no reason need be provided to the court; such privileged rejections of a venire member are limited in number for both lawyers (in Canada a maximum of 20 such challenges per side per defendant are allowed [[Government of Canada \(1985\)](#)])

- (c) The venire member is accepted into the jury, and so becomes a juror
- iv.) Steps i-iii are repeated until the prosecution and defence fail to reject the desired number of jurors.

As mentioned above, the details in this process can vary greatly by region. One of the greatest sources of this variation is the creation of jury rolls. In the United States the method is somewhat homogeneous: they are typically selected using lists of registered voters (see [Van Dyke \(1977\)](#) chapter two and [Hans and Vidmar \(1986\)](#) page 53), but in Canada their creation is far more varied. Ontario uses a combination of municipal voter lists and First Nations band lists (see [Ministry of the Attorney General of Ontario \(2018\)](#)), while in Saskatchewan - the province of the Gerald Stanley trial - the jury roll is created from the data in the central government health insurance agency in accordance with [Government of Saskatchewan \(1998\)](#).

Clearly, the variation in these methods will create differences in the universe of the sampled jury rolls relative to the population they are meant to reflect. Such differences are no doubt important to the coverage of the population which is present in the jury selection process (see [Iacobucci \(2013\)](#)), but these differences are not of primary interest to this paper. Rather, the steps presented afterwards are those to be investigated.

This leads to the two presentation methods presented in step ii, [Ford \(2010\)](#) and [Van Dyke \(1977\)](#) both note that the predominant method in the United States and Canada is the sequential-selection system. This is perhaps due to the relative efficiency of the method, as it is clear that in the sequential system voir dire need not be performed on the entire venire, only a subset. Contrast this with the struck-jury system, where the entire venire must be reviewed in every trial.

Finally, the scope of voir dire is radically different in the United States and much of the British Commonwealth. [Van Dyke \(1977\)](#) notes on page 143 that Canada and the United Kingdom do not allow questions in areas of “non-specific” bias, or bias which is not directly related to the case before the court. That is to say, while it would be perfectly valid to ask a venire member for a murder case about their work history in the United States, such a question would only be allowed in Canada or the United Kingdom if occupation was specifically related to the crime.

This difference in procedure creates a far greater emphasis on the voir dire process in the United States, as noted by [Hans and Vidmar \(1986\)](#). [Hans and Vidmar](#) go further than this, and surmise that the key reason for this marked departure in procedure is a difference in philosophy. To borrow a quote from page 63 of [Hans and Vidmar \(1986\)](#):

In Canada... the courts have said that we must start with an initial presumption that “a juror will perform his duties in accordance with his oath”

This doctrine places a responsibility on the jurors themselves to overcome their biases and accept arguments in spite of them. This stands in stark contrast to the American attitude implied by the emphasis on expansive voir dire: that certain prejudice cannot be overcome by jurors themselves. The public statements of the Stanley trial critics indicate that they subscribe to this viewpoint more than to the Canadian philosophy.

2.2 The Role of the Jury

Such a difference in viewpoint is especially relevant given the purpose of the jury. The central function of a jury in a jury trial system is to judge the innocence or guilt of an accused in light of the presented evidence, a function which has had drastically different forms throughout history. In the distant past, [von Moschzisker \(1921\)](#) and [Hoffman \(1997\)](#) report that the central function of the jury was to collect evidence, essentially assuming the role commonly performed today by police detectives. Such a role justified the archaic practice of selecting only the most “trustworthy” individuals of some renown.

This is contrasted by the modern jury, which performs no collection of evidence. It is, ideally, a panel of peers or “equals” of the accused sampled at random from the population, an idea which did not develop until 19th century Britain (see page 28 of [Hans and Vidmar \(1986\)](#)) and was not applied using random sampling until some time later (see [Hoffman \(1997\)](#), page 29 of [Hans and Vidmar \(1986\)](#), and page 16 of [Van Dyke \(1977\)](#)). The modern jury is meant to apply the law, as told to them by the judge¹, to the case at hand. Evidence of the guilt of the accused is presented to the jury by the prosecutor, while evidence meant to exonerate is presented by the defence.

The jury listens to the evidence, considers the law as presented by the judge, and must (typically) reach a unanimous decision of guilt or acquittal. Such a decision cannot be overturned by the judge of the court, and the judge must then determine sentencing based on the decision of the jury and the letter of the law¹. It should be clear that the jury therefore has tremendous power in the judgement of any case. The philosophical and ethical justification for such power is well explained by [Woolley \(2018\)](#), and best summarized by a quote from [Supreme Court of Canada \(1991\)](#):

The jury, through its collective decision making, is an excellent fact finder; due to its representative character, it acts as the conscience of the community; the jury can act as the final bulwark against oppressive laws or their enforcement; it provides a means whereby the public increases its knowledge of the criminal justice system and it increases, through the involvement of the public, societal trust in the system as a whole.

While such enthusiastic support for juries has not been expressed by all countries which practice them, the justification is entirely consistent with the histories and discussions presented by [Hoffman \(1997\)](#), [von Moschzisker \(1921\)](#), [Hans and Vidmar \(1986\)](#), [Van Dyke \(1977\)](#), and others. This suggests that the [Supreme Court of Canada \(1991\)](#) lionization of the jury system is a fair representation of the perceived role of the jury throughout those countries which use them.

2.3 Modern Peremptory Challenge Controversy

If the general utility and importance of the jury is clear, the same cannot be said for peremptory challenges. The privileged removal of a venire member² without any justification has seen persistent allegations of abuse, often around the use of these challenges by

¹[Hans and Vidmar \(1986\)](#) note that this system actually varies throughout the US, though the jury and judge powers described here are consistent across Canada.

²To be replaced by another, *randomly selected* venire member

state prosecutors.

In the United States, the criticism has focused on racial discrimination, and has led to significant changes in their allowed use, through cases such as *Swain v. Alabama* ([Supreme Court of the United States \(1965\)](#)) and *Batson v. Kentucky* ([Supreme Court of the United States \(1986\)](#)). The first of these cases, *Swain v. Alabama*, established in 1965 that the systematic exclusion of venire members of a particular race would be unconstitutional discrimination under the Fourteenth Amendment, but argued that a “*prima facie*” (or “based on first impression”) argument of discrimination was not adequate to prove this³. This placed a significant burden on the side taking issue with a particular peremptory challenge to demonstrate that the choice had been discriminatory.

However, this ruling was overturned only 21 years later in the 1986 case *Batson v. Kentucky*, which allowed the party objecting to a challenge to use a *prima facie* argument which must be countered by a race-neutral reason that satisfies the judge. If no such reason can be supplied, the challenge would not be allowed. This created a new challenge which could be used to keep a venire member despite the use of a peremptory challenge: the so-called “Batson Challenge”. While the effectiveness of this system of additional challenges is questionable both practically and in abstract (see [Page \(2005\)](#) and [Morehead \(1994\)](#), and a particularly strong response in [Hoffman \(1997\)](#)), it has only been extended to allow Batson challenges for both the sex and race of venire members⁴.

In Canada, the controversy has also had a racial component. Racial bias in Manitoba against First Nations venire members was alleged in 1991 in a report produced after an inquiry by the provincial government (see [Roach \(2018\)](#)). More damning still was the [Iacobucci Report](#) on First Nations representation in juries. This report proposed an explicit restriction to the practice when it recommended:

an amendment to the Criminal Code that would prevent the use of peremptory challenges to discriminate against First Nations people serving on juries.

Despite these recommendations and allegations, there had not been a significant political effort to reform the peremptory challenge system until the Gerald Stanley trial culminated in the tabling of Bill C75 [42nd Parliament of Canada \(2018b\)](#), which would abolish the peremptory challenge outright. As of the writing of this paper, the bill has not been approved by the Government of Canada, but it seems likely to become law in the near future.

In doing so Canada would join the United Kingdom. Significant controversy around the use peremptory challenges in the United Kingdom has already resulted in the abolition of the practice by the Criminal Justice Act of 1988. The specific controversy was the result of the Cyprus spy case in the late 1970s, which led to a “sustained campaign in Parliament and in the press alleging that defence counsel were systematically abusing it” (see [Hoffman \(1997\)](#))⁵.

³In the actual case, not a single black juror had sat in Kentucky in the previous 15 years, despite composing 26% of the jury-eligible population. In Swain’s trial, six of the eight black venire members were rejected by state prosecutor peremptory challenges, and the other two removed for cause, leaving not a single black juror to judge Swain, a black man. This was the *prima facie* argument presented by Swain’s defence team against the state prosecutors of Alabama, and it was rejected as insufficient to prove discrimination

⁴The use of Batson Challenges for sex was established in [Supreme Court of the United States \(1993\)](#)

⁵It should be noted that this did not abolish the use of “standing-aside” by the Crown, although the practice was restricted to national security trials and heavily curtailed, with strict guidelines to its use

2.4 The Role of the Peremptory Challenge

Despite these legal changes, recommendations, and a great deal of articles providing analysis against the practice (see, for example, [Hoffman \(1997\)](#)), the topic of the peremptory challenge remains controversial. The modern motivation and justification for the practice in spite of all of the controversy was perhaps best described by Justice Byron R. White in [Supreme Court of the United States \(1965\)](#):

The function of the challenge is not only to eliminate extremes of partiality on both sides, but to assure the parties that the jurors before whom they try the case will decide on the basis of the evidence placed before them, and not otherwise. In this way, the peremptory satisfies the rule that, “to perform its high function in the best way, justice must satisfy the appearance of justice.”

Such a justification is reminiscent of the now famous words of Lord Chief Justice Hewart in *R. v. Sussex Justices* in 1924: “Justice should not only be done, but should manifestly and undoubtedly be seen to be done” (as reported in [Richardson Oakes and Davies \(2016\)](#)). While these words originally only referred to the pecuniary interest of court staff involved in the case, they have since come to express the idealized expectation that both the defence and prosecution find the judge and jury acceptable, as explored by [Richardson Oakes and Davies \(2016\)](#)⁶.

This defence suggests two modern justifications for the peremptory challenge. The first is that of removing venire members with “extreme” bias, and the second is the creation of a jury which is composed of jurors mutually acceptable to both the defense and the prosecution. Those who defended the practice of peremptory challenges in Canada after the Gerald Stanley trial, including [Hasan \(2018\)](#) and [Macnab \(2018\)](#), seem to use this defence or some variant of it to argue in favour of keeping the practice. However philosophically appealing these two claims are, in light of all of the controversy surrounding the peremptory challenge, perhaps a critical and empirical examination of these assertions is warranted.

2.5 History

Such an analysis most appropriately begins with a historical explanation of the peremptory challenge. Roughly, the presentation of the history of jury trials here follows the comprehensive and exhaustively referenced description provided by [Hoffman \(1997\)](#). Two of the references [Hoffman](#) uses extensively, [Hans and Vidmar \(1986\)](#) and [Van Dyke \(1977\)](#), provided useful context while specific details provided by [von Moschzisker \(1921\)](#), [Forsyth \(1994\)](#), [Brown, McGuire, and Winters \(1978\)](#), and [Brown \(2000\)](#) helped to create a clearer picture of particular periods of jury history. Information regarding the history of the Canadian system was provided by [Brown \(2000\)](#) and [Petersen \(1993\)](#). For an excellent exploration of the nineteenth century, a formative time for the development of challenge law, see [Brown \(2000\)](#).

outlined by [Attorney General’s Office of the United Kingdom \(2012\)](#).

⁶Such grand generalizations and myth-making can also be seen in the common belief that the right to a trial by jury was originally established in the Magna Carta, an idea which is not supported by the relevant historical evidence (see [Hoffman \(1997\)](#) and [Van Dyke \(1977\)](#) for a detailed discussion and more accurate history).

It must be noted that certain important trials in the development of the peremptory challenge system have been excluded from the summary provided here. This was done deliberately, as the history presented here is only meant to explore the practice of peremptory challenges throughout history in broad terms. All of the sources listed above are much more thorough, by merit of their singular focus on the analysis of the practice from a legal and historical perspective, while this work devotes more to empirical and statistical analysis.

2.5.1 Pre-English History

Although precise timelines are hard to establish, there is evidence that jury trials have occurred in some form or another since antiquity. The concept, that of judgement by a group of peers, is so ancient that it is prevalent not only in historical records, but in myth. As [Hoffman \(1997\)](#) indicates, both Norse and Greek mythology feature groups of individuals assessing the guilt or collecting evidence about the actions of a peer.

Outside of the realm of myth, [Hoffman \(1997\)](#) reports that there is evidence of the use of juries in Ancient Egypt, Mycenae, Druid England, Greece, Rome, Viking Scandanavia, the Holy Roman Empire, and Saracen Jerusalem. It should be noted that in none of these areas was the jury trial the primary form of conflict resolution practiced. Nonetheless, it is clear the jury trial has a broad and long history of use.

Something similar to the modern peremptory challenge does not appear until Rome, however. The Roman *Judices* were groups of senators selected to judge the guilt of the accused in a legal case. According to [Hoffman \(1997\)](#), 81 Senators would be chosen to sit on one of these *Judices*, after which the litigants were permitted to remove fifteen of these Senators each. This egalitarian reduction of the jury size seems analogous to the modern peremptory challenge system, as it places the power of removal with the litigant and suggests no justification is necessary for their removal.

2.5.2 In English Law (1066–1988)

Peremptory challenge did not reach its modern form, as outlined in [2.1](#), until it was established in the English legal system. It should be noted that despite some previous debate on the topic, the most modern historical evidence suggests that the basis of the English practice was not related to the system used in the selection of *Judices* in Rome. The English system appears to be its own beast entirely.

The dominant historical interpretation is presented by [von Moschzisker \(1921\)](#) and [Hoffman \(1997\)](#): that the jury system was introduced to England during the Norman conquest of 1066 by William the Conqueror. The practice, however, was not made official until the Assize of Clarendon in 1166 by Henry II, and it was not until the outlaw of trials by ordeal (the most common method of trial at that time) in 1215, that peremptory challenges began to appear in England in the late thirteenth century. The challenges were officially recognized in 1305 when Parliament outlawed their use by the Crown, only to replace them with an analogous system of so-called “standing-aside”⁷.

⁷For a detailed explanation of this system see [Hoffman \(1997\)](#) and [Brown \(2000\)](#)

It should be noted here that although the challenges issued between the Assize of Clarendon and this 1305 act are called “peremptory,” they may not have served the same purpose, nor shared the same justification, as the modern challenges. Indeed, as [Hoffman \(1997\)](#) argues convincingly, these challenges may have been closer to modern challenges with cause. The argument hinges on the paradigm of royal infallibility and absolutism which was present in the late medieval period when the peremptory challenge first appeared (see [Burgess \(1992\)](#)).

Under royal absolutism and infallibility the argument for peremptory challenges is quite simple. If the king cannot be wrong in his judgement and he has some reason to feel that a venire member cannot serve on the jury, then he need not say why he thinks that is so, as his judgement is correct in any case. Indeed, asking for an explanation would be disrespectful and providing one undignified. The Crown prosecutors, as representatives of the king, would be similarly shielded from criticism.

Such an argument is further supported by the abolition of their royal use in 1305, the language of which suggests that peremptory challenges were originally the privilege of the Crown (see [Hoffman \(1997\)](#) and [Van Dyke \(1977\)](#)), with none being granted to the defence. [Hoffman \(1997\)](#) suggests that as royal infallibility grew out of favour, peremptory challenges were granted to the defence rather than being removed entirely.

Whatever the logic of the expansion of these challenges to the defence, their legal limits are recorded more precisely⁸. From a maximum of 35 challenges allowed at their peak in the fourteenth century, the number of challenges allowed only decreased over time until their abolition in 1988 (discussed in [2.4](#)).

2.5.3 In American Law (ca. 1700–1986)

[von Moschzisker \(1921\)](#), [Hoffman \(1997\)](#), and [Van Dyke \(1977\)](#) all agree that the early English colonists that came to North America accepted the jury system with peremptory challenges as common law well before the establishment of the United States of America. [Hans and Vidmar \(1986\)](#) note, however, that the difficulty of ocean travel and the overall indifference of appointed Crown representatives in the colonies led to an increased importance of the jury trial and the role of challenges to these early colonists as a way to exercise some degree of community control in the face of laws drafted in a distant country and implemented by unsympathetic authorities⁹.

It is somewhat interesting then, that the United States constitution makes no mention of the practice of peremptory challenges. The Sixth and Seventh Amendments specify a great deal of the jury system, including the right to public defense and an impartial jury drawn from the district of the crime, but make no mention of a right to the exercise of peremptory challenges, or any challenges whatsoever (see [Constitution of the United States \(1788\)](#)).

⁸see [Brown \(2000\)](#) for a detailed examination of the case law developing around challenges in the nineteenth century

⁹For more detail on this development among the early colonists, it is instructive to read about the Zenger trial of 1734 (described on pages 33-35 of [Hans and Vidmar \(1986\)](#)). Not only does this trial say a great deal about the attitudes of the colonists at the time, but it also presents the idea of a jury assessing guilt and “wrongness” using their own conscience rather than just settling fact. The precept of the modern jury trial in Canada (see [Woolley \(2018\)](#)) is based on this very idea

As [Hans and Vidmar \(1986\)](#) report on page 37, an original draft of the Sixth Amendment expressly included challenges for cause, but the debate around their inclusion resulted in the removal of their mention. They continue to say that at the time, even some proponents of the challenge considered the reference unnecessary, as the practice was implied by the text which remained, referring to a trial by an “impartial” jury. Another result of these debates was the adoption of the extensive voir dire process which allows questions of general bias¹⁰.

Critically, there appears to have been no discussion around the inclusion of peremptory challenges (see [Hans and Vidmar \(1986\)](#) and [Hoffman \(1997\)](#)). Despite the clear importance of the jury trial to the drafters of these amendments, it would seem the peremptory challenge was not considered to have anywhere near the same significance as judgement by an impartial jury of local peers¹¹.

Regardless of this, as [Brown \(2000\)](#) notes, the importance and use of challenges increased in the United States in the nineteenth century following American independence due to a desire to prevent the tyranny of the state. This desire also led to the adoption of a limited number of peremptory challenges for the prosecution, rather than the possibly unlimited stand-asides that were allowed under British law to prosecutors (see [Van Dyke \(1977\)](#), page 150).

While the specific numbers of peremptory challenges allowed to both sides and the required motivation of challenges for cause have varied over time (see [Hoffman \(1997\)](#) and [Brown \(2000\)](#)), they have remained a feature of the American legal system, and numerous Supreme court cases (detailed by [Hoffman \(1997\)](#)) have merely served to make the use of challenges more specific and codified. It was not until *Batson v. Kentucky* in 1986 that this system of challenges was drastically changed with the introduction of Batson challenges (described in 2.3).

2.5.4 In Canadian Law (ca 1800–2018)

Canadian law, inspired by a close relationship to both the British Crown and the United States, seems to have adopted elements of both legal systems in its development of peremptory challenges in the nineteenth century. As discussed by [Brown \(2000\)](#), Canada adopted the American practice of replacing prosecutorial stand-asides in favour of a more egalitarian limited number of peremptory challenges to both sides. Despite this, the Canadian voir dire process remains limited and much more similar to the British one, as does the system of challenges for cause (see page 48 of [Hans and Vidmar \(1986\)](#)).

One perfect demonstration of this departure is the Canadian constitution. As in the United States, the Canadian constitution fails to mention challenges. The British North America Act of 1867 (see [Constitution of Canada \(1982\)](#)), which established Canada’s independence from England, makes no mention of legal rights of the accused, indicating a deference to legal precedent in England. It is not until the Charter of Rights and Freedoms

¹⁰This is described on page 37-38 of [Hans and Vidmar \(1986\)](#), though [Brown \(2000\)](#) notes that 1807 Burr trial was also highly significant in the development of general voir dire in the United States

¹¹Indeed, as *Batson v. Kentucky* and *Swain v. Alabama* have both shown ([Supreme Court of the United States \(1986\)](#) and [Supreme Court of the United States \(1965\)](#)), the modern interpretation of “impartial” may preclude the use of peremptory challenges altogether

in 1982¹² that such rights were guaranteed in a legal Canadian document. Notably, its language is considerably more vague than the United States Sixth and Seventh Amendments, guaranteeing only “the benefit of trial by jury” (see [Constitution of Canada \(1982\)](#)).

This “eclectic” incorporation of both American and English case law, to borrow the term used by [Brown \(2000\)](#), led to a system somewhere between the English and American systems, but decidedly closer in operation to the English system. It should be noted, however, that as Canada grew more populous in the twentieth century and developed a greater legal precedent and more experienced judges of its own, this reliance upon its former colonial master and its more powerful southern neighbour seems to have diminished in importance. Viewing Supreme court rulings from recent decades reveals a great deal of reference to Canadian legal precedent rather than to the precedent of the other two countries.

2.6 Summary

The peremptory challenge, a practice of much controversy in the English-speaking world, seems to have started in its modern form as a privilege of the King of England in the thirteenth century. After its conception, it spread with English conquest and colonization, with new colonies and local governments accepting the practice based primarily on the adoption of English legal precedent. Though it was abolished in England in 1988, it remains a fixture of American jury trials, and is accompanied there by a thorough and invasive voir dire process which is not seen in Canada nor the United Kingdom.

Though the practice has historical longevity, it is not guaranteed by the constitutions of Canada or the United States, and has been a practice of considerable legal debate and significant change throughout its history. In England this culminated in the Cyprus spy trial, in the United States in *Batson v. Kentucky* and *Swain v. Alabama*, and in Canada in *R. v. Stanley*: the Gerald Stanley murder trial. As a consequence, the broad agreement of the importance and propriety of a jury has conferred little consensus on the place peremptory challenges in the selection of juries.

Indeed, it seems increasingly impossible for the jury to function in a way consistent with its demanding ideals with the peremptory challenge still present. Its spotted history and use to exclude certain minorities may undermine its purported use as a tool to ensure the acceptance of a trial’s outcome by both litigants. The three court cases mentioned above are a demonstration how the peremptory challenge can be used to create a jury which is actually unacceptable to one litigant in the case.

¹²This was the year of patriation of the Canadian constitution. As independence was granted by the British Parliament, the British North America Act outlining Canada’s laws was a British law and changing it was the prerogative of the British Parliament rather than the Canadian one. It was not until the Constitution Act of 1982 that the Canadian constitution became a Canadian law. For a more detailed history see [Sheppard \(2018\)](#)

Chapter 3

Data

Without data, performing an analysis that incorporated more than the history and legal argumentation presented in Chapter 2 is impossible. This proved problematic. While the motivation of this text was a Canadian case, no comprehensive Canadian data sets which examined jury selection in Canada could be found. The increased prominence of the jury selection process in the United States garnered a more fruitful search.

The author is heavily indebted to [Wright et al.](#); [Grosso and O'Brien](#); and [Baldus et al.](#). These authors shared their data freely with the author, providing him with a wealth of data to analyse empirically. As a consequence of the multiple separate data sets, however, care must be taken to describe each of the data sets separately in order to capture adequately the different methodologies and sources they represent. Critically, it should be noted that each of these papers represents effort on the part of the authors. As [Wright et al. \(2018\)](#) notes:

limited public access to court data reinforces the single-case focus of the legal doctrines related to jury selection. Poor access to records is the single largest reason why jury selection cannot break out of the litigato's framework to become a normal topic for political debate

Currently, the collection of jury data is difficult, as many courtrooms have not digitized past records and concerns over privacy limit the release of those records, which are stored as paper documents in the case file (see [Wright et al. \(2018\)](#)). This limits the ability of an individual to ask for summaries across numerous trials or to view the jury selection process on a scale beyond the basis of one case. Thus, to gather aggregate data the authors of these papers necessarily used different collection techniques dictated by the scope of collection desired and the procedures of the court systems from which data was collected.

3.1 Jury Sunshine Project

3.1.1 Methodology

The Jury Sunshine Project ([Wright et al. \(2018\)](#)), so named as it was carried out in order to shed light on the jury selection process, is the most extensive data set which was provided to the author. It endeavoured to collect jury data for all felony trial cases in

North Carolina in the year 2011, which ultimately resulted in a data set that detailed the simple demographic characteristics and trial information of 29,624 individuals summoned for jury duty in 1,306 trials. Note that not all entries were complete.

Due to the scope of the project, there are a number of problems which had to be solved by the authors. The first of these was simply identifying which court cases went to trial in 2011, in order to direct resources effectively. This was accomplished by downloading publicly available case data from the North Carolina Administrative Office of the Courts (NCAOC)¹ and determining the case numbers and counties of cases which went to trial. Wright et al. state that this likely missed some cases which went to trial, but that they were confident that a “strong majority” of trials was collected, which did not systematically differ from those excluded.

This list was then used to perform a pilot study to refine recording practices before undertaking a more general survey where “law students, law librarians, and undergraduate students” (called *collectors* for convenience) visited court clerk offices to collect the relevant case data, including the presiding judge, prosecutor, defence lawyer, defendant, venire members, charges, verdict, and sentence. The case files also included data about whether a venire member was removed by cause or peremptorily, and the party which challenged in the peremptory case. Using public voter databases, bar admission records, and judge appointment records, these collectors were able to determine demographic (race, gender, and date of birth) and political affiliation data for the venire members, lawyers, defendants, and judges. This data set was stored in a relational database provided to the author by Dr. Ronald Wright.

The analysis of the data provided in Wright et al. (2018) was limited to aggregate summaries of the trends at the venire member level. That is to say, they examined the strike trends for both the defence and the prosecution, conditioning on some additional variables. There was also spatial analysis performed, where different urban counties were directly compared. These analyses were not statistical in nature, and were displayed using contingency tables. Regardless the stark differences between prosecution and defence with regards to race were a key finding when the aggregate data was analyzed.

3.1.2 Cleaning

Flattening the Data

For greater expediency of analysis, the relational database of the Jury Sunshine Data was first flattened. The relational database was read into Microsoft Excel and the `readxl` package (Wickham and Bryan (2018)) was used to read the excel file into the programming language R. A wrapper for the `merge` function was developed which provided simple output detailing failed matches in an outer join in order to ensure that the flattening of the data into a matrix did not miss important data due to partial incompleteness. The code for this wrapper can be seen in D.1.

¹The link provided in the Jury Sunshine Paper to the specific source (http://www.nccourts.org/Citizens/SRPlanning/Statistics/CARports_fy16-17.asp) does not appear to be working as of January 2019, however the NCAOC seems to provide an API functionality at <https://data.nccourts.gov/api/v1/console/datasets/1.0/search/>

This wrapper revealed only a small number of irregularities in the data, which are detailed in [B.1](#):

- i.) Twenty-nine charges missing trial information such as the presiding judge (all of trials with IDs of the form 710-0XX)
- ii.) Twenty-six prosecutors not associated with any trials and missing demographic data
- iii.) One trial missing charge information

Ultimately, the jurors for trial ID 710-01, the trial missing a charge from above, were included in the data as their records were complete otherwise. The prosecutors and charges which could not be joined were excluded from any future analysis, as they could have easily been included by collectors by accident. Due to the small relative size of these inconsistencies relative to the size of the data set, they did not cause concern.

Uninformative Columns

Of course there were other irregularities in the data than the obvious ones that arose in the flattening process. There were a handful of likely sources for these errors. The first of these is the anonymization of the data for public use. The private data includes a wealth of privileged data such as juror name and address, and these were removed in the data given to the author.

As a consequence of this anonymization as well as the inclusion of rarely used columns such as those for additional notes, some columns of the data contained only NA values. Most baffling of these was the `BirthDate` variable in the `Jurors` table, as there was no clear reason for this data to be missing. Thankfully, none of the missing columns were relevant to the joins performed in flattening, and they would have been only secondary in data analysis. As a consequence, these uninformative columns were simply removed from the data.

Coding Inconsistencies

Related to this problem was the issue of inconsistently coded variable levels. An example of these inconsistencies would be levels recorded as both lower and upper case letters, or the presence of “?” instead of “U” for unknown values. It is very likely this inconsistency was a direct result of the data collection method which used many data collectors working independently in different places at different times. Thankfully, [Wright et al.](#) provided the codebook used by data collectors, which served as the authoritative reference for the admissible factor levels of demographic variables. Solving this problem was as simple as setting all demographic variable levels to be uppercase and replacing obviously mis-specified levels.

One specific inconsistency which should be noted is that of the outcome, which had a handful of entries recorded as “HC”, an inadmissible level not defined by the codebook. It is quite possible that this level represented a typo, as the “H” and “G” keys are adjacent on the American QWERTY keyboard layout, and “GC” was the code for ‘guilty as charged’, but out of a desire to be conservative with the data the occurrence of this outcome was replaced with U instead. Additionally, the inadmissible level “G” was replaced by GC.

Swaps

A more difficult problem of level misspecification was the presence of what appeared to be columns with swapped values, frequently occurring with the gender column (the admissible levels of which are “M”, “F”, and “U”) and the political affiliation column (the admissible levels of which are “D”, “R”, “I”, or “U”). The aforementioned “swaps” appeared as records in which, for example, the gender was recorded as “R” and political affiliation as “M”. More complicated swaps of three columns also occurred. To address this problem, the `IdentifySwap` function was written (see line 108 in [D.1](#)).

The `IdentifySwap` function accepts two arguments: a data frame with named columns and a named list of vectors of the acceptable levels for some of the column names. It then performs vectorized checks of the specified column names and presents any rows which may have swaps or errors interactively to the user, along with a suggested reorder to “un-swap” the row. The user can press enter to accept the suggested reordering, enter some other reordering, or enter 0 to indicate that the row was not a true swap, but simply an error. The un-swapped entries are then returned to the data, and the rows with errors have the erroneous values replaced by “U”, the universal code for “Unknown” within all data variables².

The source of these swaps is also most likely the data collection method. The codebook provided specifically notes that the data collection was meant to record the race (R), gender (G), and political affiliation (P) data in the form RGP, but it is not inconceivable that it would occasionally have been recorded or entered in some other ordering in the tedium of data entry. In any case, this problem affected only 431 records of the nearly 30,000, suggesting that the recorded error rate was not unacceptably large.

Charge Classification

Perhaps the least regular data in this data set was that of the charge text. Due to the lack of any codebook guidance about the standard way of recording a charge in a trial, identical charges were recorded in numerous ways. The first method used to combat this was removing non-alphanumeric characters, extra spaces, and converting all charges to lower case. This still left a considerable variation, however. Consider the charge of breaking and entering, for example, even with this simple preprocessing performed the entries varied significantly (e.g. “break or enter”, “breaking andor entering”, “breaking and or entering”, etc.).

As a consequence, the processing was more involved. First the most common versions of the charge text for the charges were all regularized to be identical (see `StringReg` in [D.1](#)). Next, a regular expression classification tree was developed, which would also account for specific features of a charge. When identifying murder, for example, it seemed important to ensure attempted murder was separated from murder itself, and separating first and second degree was also desired. This tree would, when presented with a charge, apply the regular expressions at each node to the charge. If the charge matched the expression at a node, the regular expressions of that node’s children were applied to the charge until it was classified to some leaf node, each of which had a standardized value which replaced

²One notable exception to this insertion of “Unknown” was the case of the judge Arnold O Jones II, whose gender was not recorded in the data, but who was identifiable as a man using a quick Google search of his unique name

the charge. A small example of this structure is displayed in Figure 3.1, and the full tree is visualized in B in Figure B.1.

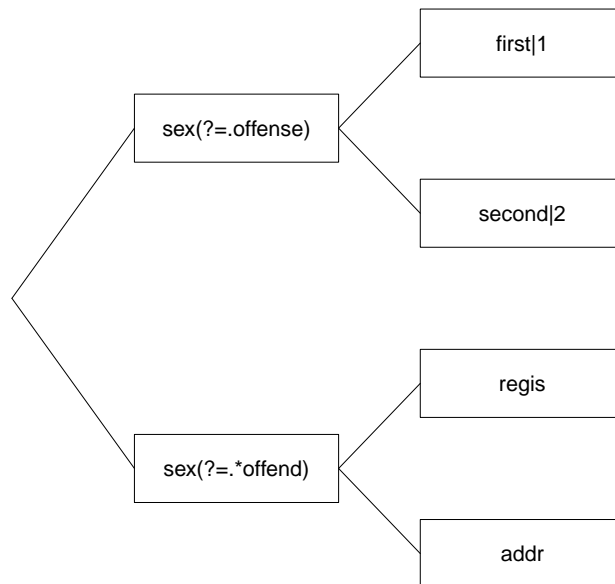


Figure 3.1: A example of a simple charge classification tree to separate the sexual offenses from charges leveled against previously known sex offenders. A charge would be classified from most general on the left to most specific on the right.

By performing regularization using this charge tree, regularized charges were guaranteed. The cost of this regularization was the inability to classify all crimes, however. Of the 1407 charges present in the data, the tree provides regularization for 1209. With additional time and inspection of the failed matches, the tree could conceivably be expanded to regularize all charges. As this was not the primary investigation of the report, however, such effort was not expended.

Instead, a number of helpful aggregation and extraction functions were developed to allow for the charges to be further simplified, as seen in D.1. In this report, they were aggregated by intuitive classes: sex-based offenses, thefts, murders, drug charges, violent offenses not otherwise classified, and driving charges. However, other classes, such as the North Carolina felony classes themselves (as provided by [North Carolina Sentencing and Policy Advisory Commission \(2017\)](#)), may provide a more informative classification rationale.

Variable Level Renaming

The final step of the data cleaning process was to convert the uninformative codes used to indicate variable values to more intuitive and clear names (for example to convert “I” in the political affiliation variable to “Ind”, a clearer indication of independent). Certain variables which were already clear, such as gender (codes “M”, “F”, “U”), were not renamed due to the clarity of the one letter representations.

3.1.3 Variable Synthesis

In order to expand the possible analysis and visualization potential, a number of variables were synthesized from the Jury Sunshine data set. They are detailed below.

Race Match A logical variable which is true for a venire member if they are the same race as the defendant, and false otherwise. This variable was motivated in particular by the Gerald Stanley trial, in which the implicit contention of those who have taken issue with peremptory challenge is that the First Nations venire members were removed by the defence as their race did not match that of Stanley in the racially charged trial.

Guilty Logical indicator indicating whether the trial outcome was guilty or not

Visible Minority Logical indicator of non-white venire member race

Race of Striking Party Factor variable which gives the race of the prosecution if the venire member was struck by the prosecution, the race of the defence if the venire member was struck by the defence

Simplified Race Due to the scarcity of the other minority races, this variable simplified the race provided to white, black, or other for the venire member

Simplified Defendant Race The same as the simplified race for the defendant races

3.2 Stubborn Legacy Data

3.2.1 Methodology

[Grosso and O'Brien \(2012\)](#) also provided data to the author, albeit a more limited set. This study, also based in North Carolina, focused on the trials of inmates on death row as of July 1, 2010, yielding a total of 173 cases. In each proceeding, the study examined only those venire members not excluded for cause, and critically the analysis of the study focused only on prosecutorial peremptory challenges. Besides collecting demographic data as in the Jury Sunshine Case, this study also collected attitudinal data for the venire members.

Staff attorneys from the Michigan State University College of Law were responsible for the data collection in this study. The work was performed similarly to the Jury Sunshine Data, using case files to collect information about the court proceedings such as the peremptory challenges used, presiding judge, prosecutor, and defence lawyer. Detailed verdict and charge information was not collected, as the preselection criteria of death row inmates made the verdict clear, and the death penalty can only be applied for certain crimes.

To collect demographic and attitudinal data, the juror questionnaire sheets were consulted³. These sheets are typically used as a component of voir dire, in order to make the process more efficient and determine venire members categorically ineligible for jury duty. As a result, they inquire about opinions on the death penalty, for example, as well as

³As [Grosso and O'Brien \(2012\)](#) observe, self-identified race may be the most accurate source of racial group identification

demographic questions. As not all jury questionnaires were available, additional information was collected from jury roll lists to determine the races of the final jury members. It should be noted that this collection was done blind and to high standards of proof, and a reliability study carried out in [Grosso and O'Brien \(2012\)](#) indicated that under this system the race coding was 97.9% accurate when the standards were met. Those for whom the standards were not met were marked as “Unknown.”

The analysis performed in this paper was more statistical than in the Jury Sunshine Data. Contingency tables generated using the data were tested using Chi-squared independence tests, and a simple logistic regression model was created to predict prosecutorial strikes. One minor criticism which could be made of their methodology is the lack of a consistent level to their tests. It seems that rather than class these tests as significant or not, these tests were simply performed to report the p-values they returned. Additionally, there are possible multiple testing issues as the study seems to indicate multiple tests were performed on each table, with the specific test used to generate the reported p-values not clearly indicated.

3.2.2 Cleaning

3.3 Philadelphia Data

3.3.1 Methodology

[Baldus et al. \(2001\)](#) presents a similar data set collected using similar means. Court files such as the juror questionnaire, voter registration, and census data were all used to complete juror demographic information for 317 venires consisting of 14,532 venire members in Philadelphia capital murder cases between 1981 and 1997⁴. It should be noted that this data included only those jurors kept or peremptorily struck, venire members struck for cause were not included in the data. The procedure used to determine race using the census and voter registration polls was quite complicated, but was rigorously performed using accepted census methods to a standard of 98% reliability⁵.

In their incredibly thorough analysis of the data, there were findings consistent with both the Jury Sunshine and Stubborn Legacy data. The defence and prosecution seem to follow mirrored patterns of racial preference in the use of peremptory challenges, even when controlling for other possible confounding effects.

3.3.2 Cleaning

One interesting quirk of the Philadelphia data set was missing values. While the codebook describing the data explicitly stated a number of variables should be recorded as binary values. In the provided data files, however, these variables were missing for a majority of the observations. In the case of the “FINLJURY” variable for example, an indicator of whether the jury member was included in the final jury, there were 4626 records with a value of 1, 3 with a value of 0, and 12890 missing values. These missing values were

⁴This study took into account the sampling error by reweighting venires based on the year of the trial and the defendant race, as court records showed that the sample coverage varied over these factors

⁵Additionally, imputation was only performed in a small minority of cases

assumed to be zero, as using this assumption created a data set which was consistent with that reported in [Baldus et al. \(2001\)](#).

Chapter 4

Empirical Analysis

With this data cleaned and processed, questions can now be posed and addressed through analysis. A few obvious questions come to mind, considering the previous work done on this subject and the modern controversy surrounding it. First, there is the obvious question of not only the possible racial imbalance of peremptory challenge use, but how this imbalance changes with the race of the defendant. In the Gerald Stanley trial, for example, the critical aspect of the trial was not the use of peremptory challenges in abstract, but how their use interacted with the race of Stanley.

Aside from these investigations, we may wonder whether the most common arguments posed in favour of peremptory challenge are satisfied in this data. As discussed in 2.4, there are two primary arguments. The first is the argument that the peremptory challenge is necessary to remove the “extremes of partiality” present in the venire for both sides, that is to remove the most extremely biased jurors. This goal is complemented by the ability of the judge to remove jurors with cause, which is also designed to remove those jurors with extreme bias. The second argument is the creation of a jury which is mutually acceptable to both parties in the trial.

4.1 Extremes of Partiality

While creating a quantitative judgement on the acceptability of a jury is somewhat difficult, measuring the extremality or abnormality of observations is a critical function of statistics. With this in mind, a very simple calculation was performed. The central claim of the advocates of the use of peremptory challenge is that it is only used to remove extreme cases of bias. If that is so, then the proportion of venire members removed by peremptory challenge should reflect this concept.

Of course, this cannot be rigorously tested, as there is no way of knowing the true distribution of bias among jurors. That does not mean nothing can be said, however. As [Nisbett and Kunda \(1985\)](#) notes, there is a tendency of people to guess that a distribution is normal when asked to guess the distribution of social attitudes¹. Additionally, math-

¹This problem is not helped by the notoriety of the normal distribution, as it is commonly the distribution used when performing tests (likely due to the utility of the Central Limit Theorem) and generating visualizations of a general distribution

Table 4.1: The implied statistical extremity bound for symmetric rejection in the datasets given different distributional assumptions

Data	Rejection Rate	Normal	Chebyshev Limit
Sunshine	0.434	0.781	1.517
Stubborn	0.659	0.442	1.232
Philadelphia	0.736	0.337	1.166

ematical constraints such as the Chebyshev inequality (see [Weisstein \(2018\)](#)) provide an upper limit to the dispersion of any distribution.

This study suggests that it would not be unreasonable to view the overall causal and peremptory challenge rates as the tails of a normal distribution, and the Chebyshev limit gives an estimate of the extremality of rejections given a maximally dispersed distribution of opinions. Table 4.1 provides a summary of the rejection rates of the different data sets and the implied standard distance from the centre that these imply for symmetric rejection.

Obviously, it is not possible to comment with authority on the presence of partiality in the population. Indeed, given the large divide that appears to be present politically in the United States and the rest of the Western world today, it may be easier to argue for a maximally spread distribution than a centralized one. Regardless of this difficulty, it is difficult to justify that 43% of a dataset is “extreme” statistically. In the normal case, this suggests that the rejection boundary is less than one standard deviation from the mean, i.e. that the typically sampled point will be too extreme. The Chebyshev limit is not much better, suggesting that the rejection boundary is at most 1.5 standard deviations from the mean in either direction.

This low rejection boundary given any distribution suggests that the peremptory challenge is not simply being used to remove “extremes of partiality.” Rather, it seems that the argument used to support and justify the practice cannot be reconciled well with the data, suggesting systemic over-use relative to its supported use. This leads naturally to the question of how exactly this legal instrument is over-used, and why.

4.2 The Impact of Race

The racially-motivated controversy surrounding peremptory provide one possible route to investigate the pattern of peremptory strike over-use. To begin, a simple marginal investigation was performed to explore the impact of the simplified race on the peremptory strike probability. The result of this investigation is displayed in Table 4.2. Of particular interest is whether any race is far more likely to be struck by peremptory challenge than the others, as this might suggest that race is the target of the over-use of strikes.

The differences in these probabilities are significant in the Sunshine data at $\alpha = 0.05$ by merit its size, but an effect size of 2% is hardly relevant if one considers that the size of each empanelled jury is typically only 12. Moreover, the average number of venire members which is interviewed to create the jury is only 19. Perhaps there is a more interesting relationship present at the racial level. Taking inspiration from *Swain v. Alabama*, *Batson v. Kentucky*, and *R. v. Stanley*, perhaps viewing the relationship between venire member

Table 4.2: The conditional probability of a venire member being struck peremptorily by the simplified venire member race across data sets. These values are smaller than the values presented in the extremity analysis as only the individuals which were identifiably removed by peremptory challenge are counted in this table. Regardless, the comparisons remain similar if the unattributed removals are included. Note that the Philadelphia trial data only indicated black and non-black venire members and so only two numbers can be reported.

Data	Black	Other	White
Sunshine	0.23	0.24	0.25
Stubborn	0.65	0.36	0.66
Philadelphia	0.67	0.68	

race and defendant race would be informative. This relationship is displayed in Figure 4.1.

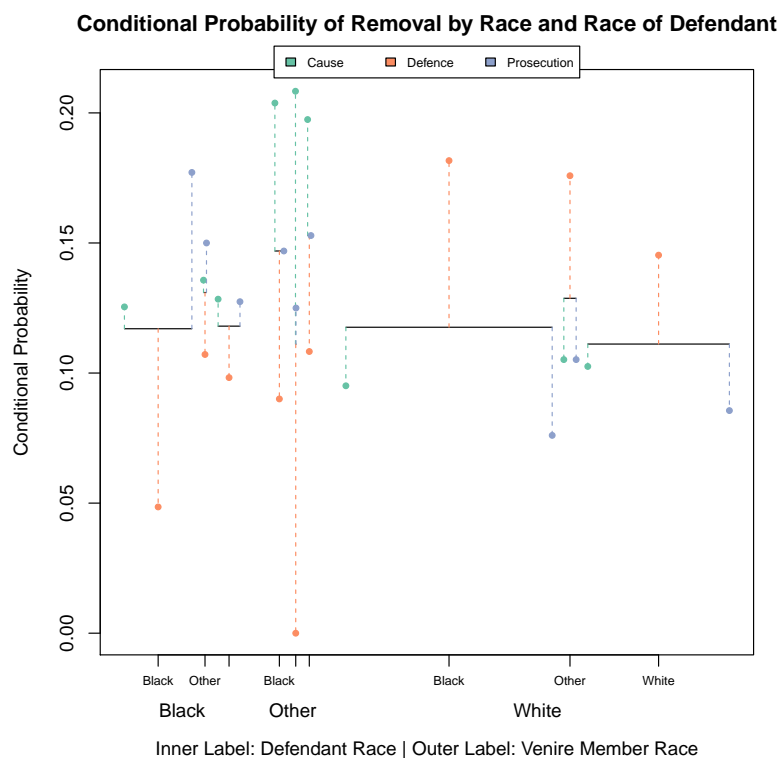


Figure 4.1: The conditional probability of successful challenges given the venire member and defendant race, with the expected value represented by the horizontal black line, and the observed values represented by the point at the end of the dotted line. Each horizontal black line corresponds to a particular venire member and defendant race combination, with a length proportional to the number of venire members with that combination. The dashed vertical lines, coloured by challenge source, start at these horizontal lines and end at points which show the observed probability of a challenge by that source for the given racial combination.

A detailed description of this plot and its development which includes a discussion of the

principles of graphics and perception which were used to devise its form is presented in [A²](#). Instead, the most interesting patterns visible in the plot will be discussed here.

First, a small explanation of the plot. The plot displays the relationship between three categorical variables: venire member race, defendant race, and disposition (whether a venire member is struck and by whom). The vertical axis corresponds to the conditional probability of the disposition given a race and defendant race combination. Racial combinations are placed along the horizontal axis, and each combination corresponds to one horizontal black line in the plotting area. The length of these lines is proportional to the number of venire members in the data with the corresponding racial combination, and their vertical positions are the mean conditional probability of a venire member being removed by a challenge for that particular combination. The dashed vertical lines, coloured by disposition, start at this mean line and extend to the observed conditional probability of the corresponding disposition for the relevant racial combination. As a consequence, this plot can be viewed as a visualization of the test of a specific hypothesis:

$$D|R_{VM}, R_D \sim Unif(\{1, 2, 3\}) \quad (4.2.0.1)$$

Where D, R_{VM}, R_D are random variables representing the disposition, venire member race, and defendant race respectively. In words: the conditional distribution of the disposition given the racial combination is uniform. This implies that all three strikes occur with the same probability for each racial combination, though they may differ between racial combinations. Such a hypothesis allows for certain racial combinations to experience a higher strike rate generally, but constrains the strike rate to be the same for all parties, which would imply that all parties in the court pursue an identical strike strategy across all venire member and defendant race combinations.

Clearly, Figure 4.1 casts some doubt on this hypothesis. While the horizontal black lines tell a very similar story to Table 4.2, with little variation between them, a number of other striking patterns are visible. The first, and most obvious of these, is the main effect of venire member race. While the aggregate removal rates do not seem to depend on the race of the venire member, it is clear that the defence and prosecution pursue radically different strategies. The defence seems biased towards a jury with more venire members from visible minorities; all orange points are below the horizontal lines for the black and other venire members, indicating these groups are less likely to be struck by the defence than expected, while the points are above the lines for the white venire members, indicating a higher than expected probability of defense removal for white venire members. The prosecution seems to mirror this tendency, striking the white venire members at a lower rate than expected and the visible minorities, especially black venire members, more often than expected.

The addition of defendant races shows another interesting trend. It would seem that the aforementioned tendencies of the prosecution and defense are strongest for black defendants, which have the strongest departure of the conditional probabilities from the expectation. The defense and prosecution seem to have slightly more similar habits when the defendant is white, despite their opposite tendencies in all cases. Finally, it would seem that the removals with cause have tendencies similar to the prosecution, as the points representing the conditional probability of a venire member being removed with cause are always on the same side of the expected line, an event which would occur with probability

²Here it suffices to mention that much of its design was motivated by the philosophy of [Tufte \(2001\)](#) and the results of [Cleveland and McGill \(1987\)](#) on the accuracy of visual perception.

$2^{-9} \approx 0.002$ under the hypothesis of independent uniform strike rates. Further exploration of the agreement of these two strike tendencies is explored in 4.5.

While Figure 4.1 is quite suggestive, the widths of certain horizontal black lines, in particular those for venire members with a race other than white or black, suggest that perhaps some of the more extreme tendencies are simply a result of the well-known higher variation of samples with small sizes. In order to see the true nature of the noted departures some incorporation of the variation one expects from each observed value is required. This is accomplished by the addition of approximate 95% multinomial confidence intervals using the `MultinomialCI` package in R, which implements simultaneous confidence intervals for multinomial proportions following the method presented in Sison and Glaz (1995). These confidence intervals can be seen in Figure 4.2.

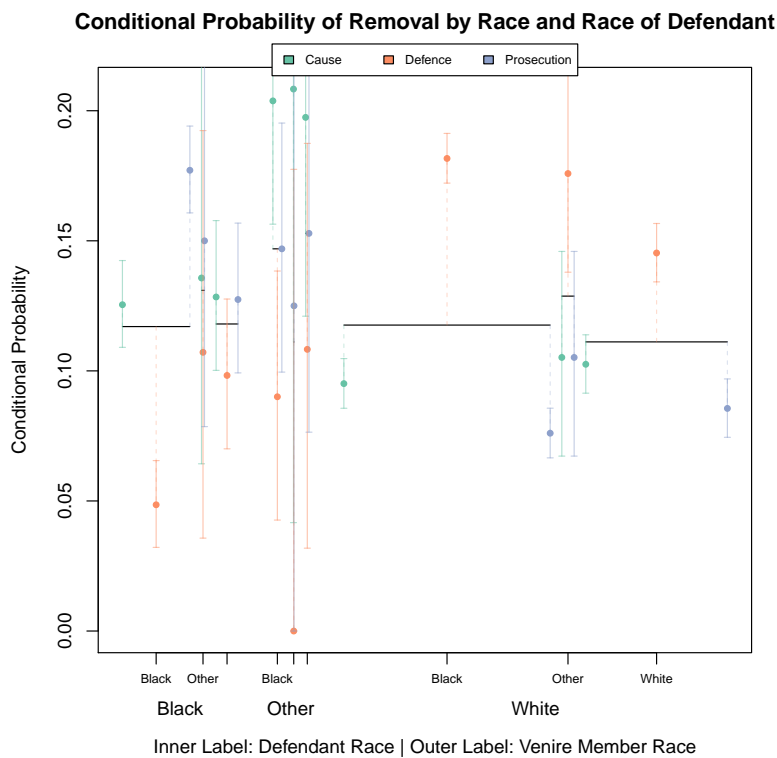


Figure 4.2: The plot of conditional strike probability by racial combination from above with confidence intervals added. Note that many of the seemingly striking departures seen are insignificant when these confidence intervals are applied.

As suspected, some of the results for the smaller sample sizes do not seem to be significant. The results for the larger groups, in particular for white venire members or black defendants, are significant, however. It should be noted that these simultaneous confidence intervals do not constitute a proper statistical test of the impact of race, they are rather a way of visually providing a viewer some sense of the expected variability in the data over repeated sampling. More rigorous testing is performed alongside the model building in 4.4.

4.3 Other Factors

Of course, it would be unfair to immediately assume that the cause of the racial patterns observed above is race itself. There may be a plethora of attitudes associated with race that could serve as legitimate cause for a peremptory challenge. As noted by Justice Byron R. White in the majority opinion in [Supreme Court of the United States \(1965\)](#)

[The peremptory challenge] is no less frequently exercised on grounds normally thought irrelevant to legal proceedings or official action, namely, the race, religion, nationality, occupation or affiliations of people summoned for jury duty. For the question a prosecutor or defense counsel must decide is not whether a juror of a particular race or nationality is in fact partial, but whether one from a different group is less likely to be.

This quote leads directly to the heart of the problem. Without detailed transcripts indicating how the venire members were questioned, it cannot be known if the aggregate pattern of removal is the result of racially based strikes, or whether the lawyers determined valid reasons for a peremptory challenge during the voir dire process. For example, if defence attorneys reasonably assumed that trust in and deference to authority and law enforcement would make a venire member predisposed to reject arguments provided about possible mishandling of evidence without proper consideration, this would be reasonable grounds for peremptory challenges of individuals with that opinion. If such opinions are distributed heterogeneously by race, the aggregate pattern may appear to reflect racially-based decision making by the defence attorneys.

[Revesz \(2016\)](#) provides, inadvertently, data which might support the above reasoning in defence of peremptory challenges in the United States. He notes that the distribution of political affiliation in the United States is not consistent across races, with black voters far more likely to vote for the Democratic Party and far less likely to vote for the Republican Party. If political affiliation is used as a surrogate for ideology and point of view, this suggests that the observed pattern could be the result of defence lawyers removing conservative venire members and prosecution lawyers attempting to remove liberal ones. As the Sunshine data has political affiliation, these aggregate results can be examined for the data used to generate Figure 4.1. Figure 4.3 displays the conditional probability of political affiliation across races and genders.

What is immediately apparent viewing this plot and the data in [Revesz \(2016\)](#) is how closely the two data sets agree. This “ideological imbalance”, as [Revesz](#) aptly calls it, is a clear confounding factor and a possible source of a legitimate cause for an initially suspect overall trend. As such, it was investigated using the mobile plot.

To control for the defendant race as well, which already appears to be important, the venire members were split into the simplified racial groups black, white, and other. Then mobile plots of the conditional strike probabilities for the different venire races given the defendant race and political affiliation were generated. Figures 4.4a, 4.4b, and 4.4c display these mobile plots.

This sequence of three plots immediately suggests that a political argument is insufficient for this data. For each venire member race and defendant race, the political affiliation of the venire member does not radically change the pattern of strikes for any party in the court. Rather, the court tendencies for each political affiliation, venire member race, and defendant race seems to follow the pattern seen in 4.1 for all political affiliations with the

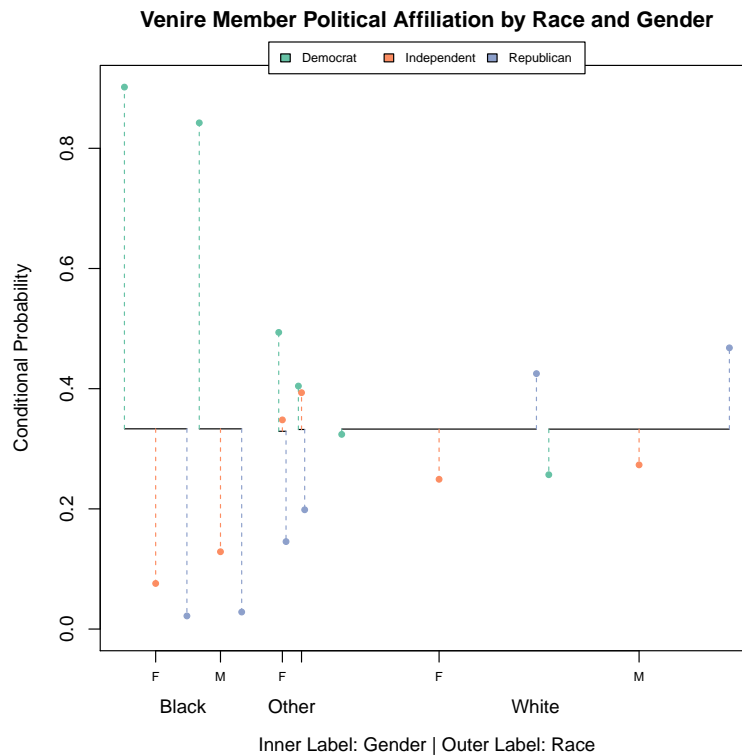


Figure 4.3: Conditional probabilities of political affiliation by race and gender. Note how the black venire members are far more homogeneous than the white venire members for both genders.

exception of some very small subgroups of black venire members³. It should be noted that many of the other, less noticeable subgroups represented by this particular set of plots will be too small to provide statistically significant results, the consistency of the pattern of strikes across political affiliations provides rather stark evidence against a possible claim of political affiliation being the reason for the strikes in the data.

Another factor which may have an impact is that of gender. While *J.E.B. v. Alabama* (Supreme Court of the United States (1993)) has also ruled peremptory challenges for this reason alone unconstitutional, it is noted in Van Dyke (1977) on page 152-153 that prosecutor's guidelines have, in the past, recommended using peremptory challenges to remove female venire members⁴, and so perhaps it is a relevant factor. Additionally, there

³The subgroups, black republican venire members for white or other race defendants, have sizes 1 and 10, respectively. The study examines a total of 29636 venire members.

⁴The guidelines for sex written by Jon Sparling, an assistant district attorney in Dallas, read specifically:

- i.) I don't like women jurors because I don't trust them.
- ii.) They do, however, make the best jurors in cases involving crimes against children.
- iii.) It is possible that their "women's intuition"[sic] can help you if you can't win your case with the facts.
- iv.) Young women too often sympathize with the Defendant; old women wearing too much make-up are usually unstable, and therefore are bad State's jurors.

If data on make-up use and jury outcome is ever collected perhaps Mr. Sparling's bold claim can be tested, but until then it is better treated as prejudice.

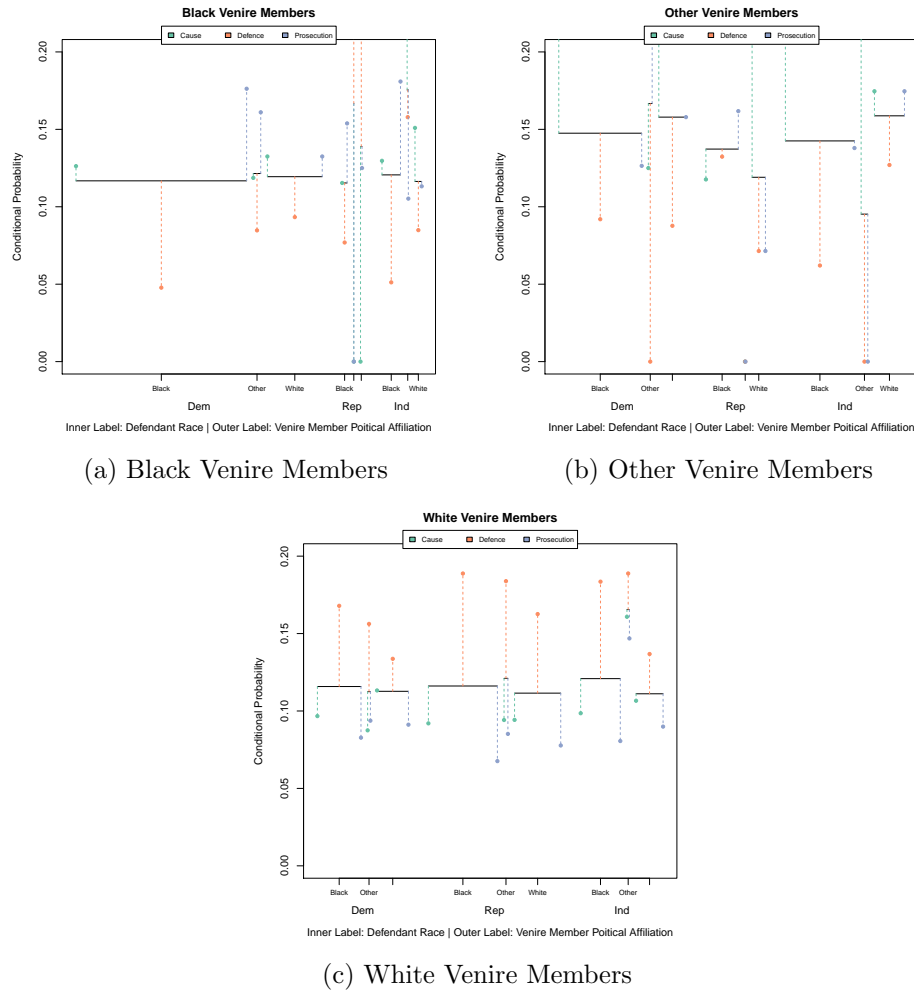


Figure 4.4: Conditional probability of venire member strike by defendant race and political affiliation, split by race. Note how the pattern of conditional probabilities is the same across political affiliations for the same defendant race within a particular venire member race, and this pattern is the aggregate pattern seen in Figure 4.1

is a relationship between gender and race in the data, as noted in [Wright et al. \(2018\)](#) black males are highly under-represented relative to black females. Luckily, the relationship between race, gender, and peremptory challenges can be visualized quite neatly without the need to split the plots as was done in Figure 4.4. Figure 4.5 displays the mobile plot for race, gender, and strike source.

The same pattern is seen here for the most part, that of a dearth of defence strikes against visible minorities and a surplus against white venire members, with a mirrored tendency for both the prosecution and challenges with cause. One may ask the question if such a switch is just a quirk of the data, and if conditioning on gender alone, for example, would create a similar pattern of switching tendencies of the court. Figure 4.6 shows this plot.

The characteristic switch is not present. At this point the message from these additional mobile plots should be clear. The dominant determinant of the strike probability for a venire member is their race. The race of the defendant does impact this somewhat, but plots across gender and political affiliation for both venire members and defendants

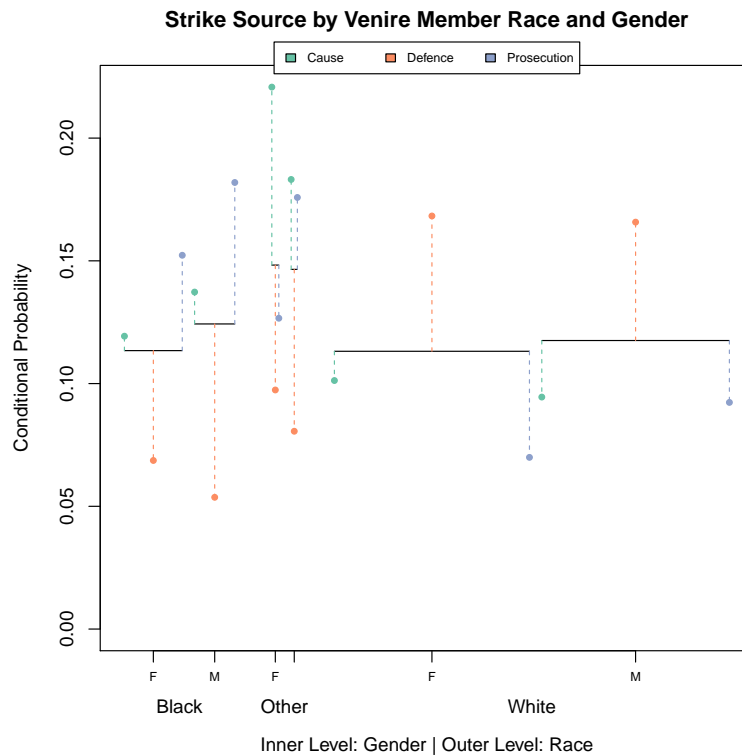


Figure 4.5: Conditional probabilities of strike source by race and gender. Note that the pattern is nearly identical for the genders within each racial group.

in numerous combinations suggest that race is the dominant factor in determining the probability a venire member will be struck by the prosecution, defense, or removed by cause. This hypothesis, suggested heavily by these plots, is examined more rigorously through the construction and testing of specific models in 4.4.

4.4 Modelling

Controlling for the myriad of possible confounding factors motivated the fitting of multiple models to test the significance and magnitude of the impact of certain variables on the probability of venire member rejection. Here, the use of the mobile plot in Figure 4.1 and others is suggestive of a model by design. The creation of a plot with conditional probabilities contingent on other factors is suggestive of a conditional multinomial distribution, and so fitting multinomial log-linear regression seemed a natural choice.

It should be noted that early modelling instead used Poisson regression, but as the conditional distribution of a particular outcome given some marginal count is multinomial (as shown in C), these two models are essentially equivalent. There are known transforms (Baker (1994)) to move between the models and their equivalencies are well documented (Lang (1996)). For greater interpretability, however, the fitting performed used multinomial models as implemented in the `nnet` package in R, which implements a method of fitting multinomial models discussed in Venables and Ripley (2002). While in Poisson models the interaction terms are the only quantities of interest, and the main effects merely

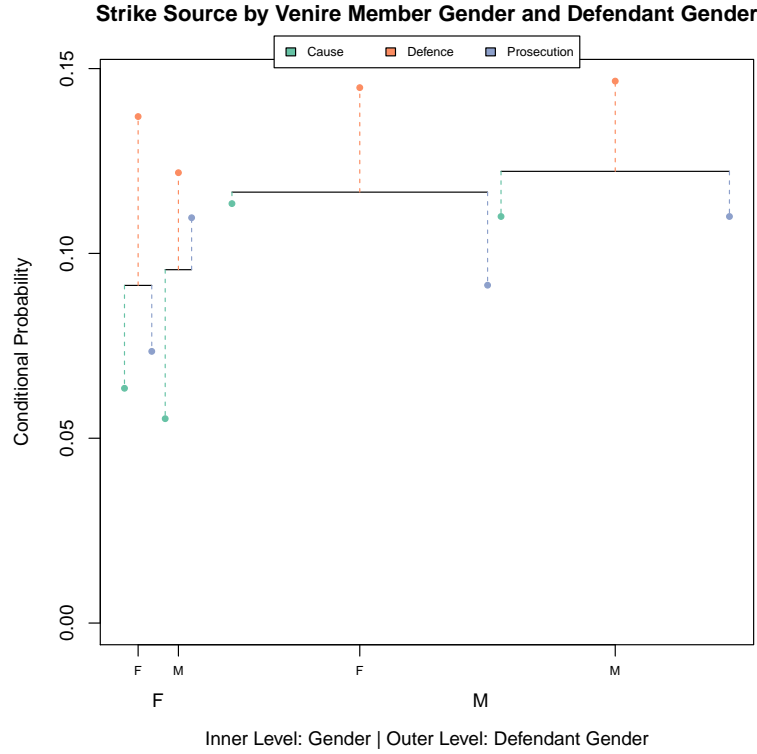


Figure 4.6: Conditional probabilities of strike source by venire member gender and defendant gender. The characteristic swap of the oppositional preferences cannot be seen here.

serve as constraints, the multinomial fit leads to the more intuitive main effect importance for the different factors.

In order to create a single model to test the statistical significance of the differences observed for strike rates by race, defendant race, and party doing the striking, a saturated poisson regression model was fit to the data. Letting i denote the level of the venire member race, j the defendant race, and k the disposition, the numbers of observed venire members in each ijk combination, y_{ijk} were modelled as Poisson-distributed random variables with expectation λ_{ijk} . A saturated model was then fit to the data, that is a model described by the equation:

$$\log E[y_{ijk}] = \mathbf{x}_{ijk}\beta = \beta_o + \beta_R x_{i..} + \beta_D x_{.j.} + \beta_S x_{..k} + \beta_{R:D} x_{i..} x_{.j.} + \beta_{R:S} x_{i..} x_{..k} + \beta_{D:S} x_{.j.} x_{..k} + \beta_{R:D:S} x_{i..} x_{.j.} x_{..k} \quad (4.4.0.1)$$

Where $x_{i..}$ indicates the race level of the ijk cell, and $x_{.j.}, x_{..k}$ are defined analogously for the defendant race and disposition. The interaction terms then serve to answer questions about the racial pattern of strikes which is utilized by each party given the defendant race. Most interesting to this investigation is the third order interaction term. This term indicates a significant difference in racial strike patterns given the party striking and the defendant race. In other words, this term accounts for different patterns for the different parties which are not independent of the defendant race.

While this second tendency seems to have no justification beyond race, the dominant tendency may have other justification than simply skin colour. As was noted by “Ideological Imbalance and Peremptory Challenge”, black individuals are more consistently aligned with the democratic party, and as a consequence a lawyer which suspects this political bias will impact the trial outcome would preferentially strike or keep black jurors in order to keep as many left wing individuals as possible. In this data, this political imbalance is incredibly prevalent, as can be seen in Figure ?? **Add the plot of this effect here, elaborate on this pattern more based on the plot.**

4.5 Prosecution and Judge Homogeneity

As noted in 4.2, the prosecution and judge seem to match in their strike tendencies for every combination. This suggests that both challenges with cause and the prosecution tend to have the same effect on the jury composition, though the magnitudes can differ greatly for these two strikes. An immediate explanation to this is offered by [Hans and Vidmar \(1986\)](#), who outline, on pages 69-70, the skill and tact required to effectively propose challenges with cause. In order to determine an individual’s bias, it is frequently the case that a direct question will fail to garner an honest response due to social pressures. As a consequence, the questions asked of venire members must be carefully presented.

Using this as a motivation, an obvious possible explanation for the challenges with cause is that the prosecution is simply more experienced on average than the defence. To determine the veracity of this claim, the licensing year of each lawyer was subtracted from the outcome date of each trial. The resulting distribution of years of experience was then plotted in back-to-back histograms as shown in Figure 4.7.

Clearly, this hypothesis is not supported by the data. It seems the typical defence lawyer is more experienced than the typical prosecutor, not less. Indeed, the prosecutors seem to be much more likely to be inexperienced than the defence lawyers.

4.6 Case Level Summary

While [Wright et al. \(2018\)](#) reported a great deal of aggregate statistics about the venire members themselves, one piece of investigation which was lacking was an analysis which aggregated and viewed the trends for the cases, rather than simply for individual venire members. As we cannot know why a potential venire member is struck individually, and viewing their aggregate statistics tells us nothing about how different strikes relate to each other, it is possible we are viewing some effect which is not a result of persistent bias across trials, but is rather the result of some other effect.

By aggregating the venire members by trial and viewing the demographic trends in strikes and behaviour at this level, we gain a more detailed insight into the impact of challenges at a more relevant scale. Additionally, such aggregation allows for the synthesis of certain measures, such as a disistributional difference via the Kullback-Leibler divergence ([Kullback and Leibler \(1951\)](#)), which would otherwise not be well defined. This particular perspective of the data has also not been explored by any other studies known to the author.

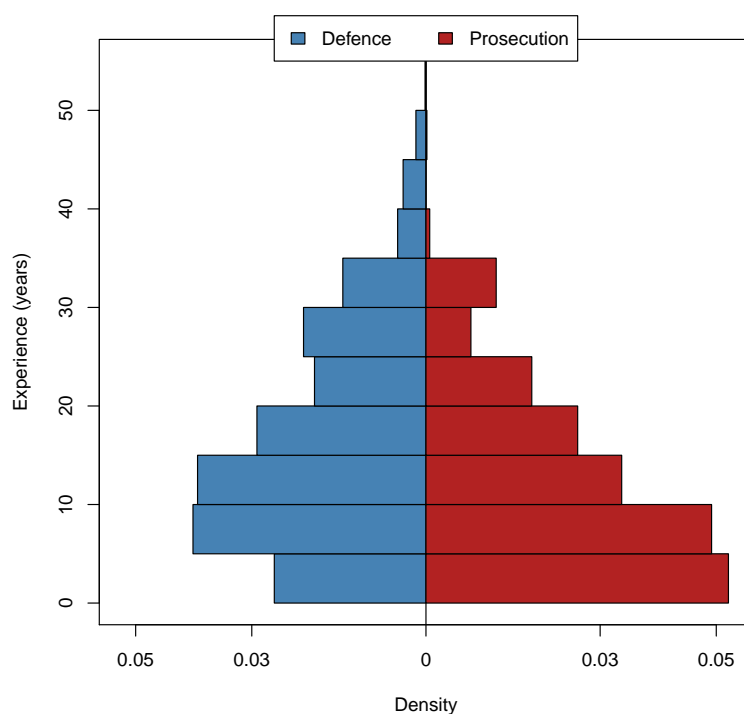


Figure 4.7: Distributions of lawyer experience for prosecutors and defence attorneys

4.7 Useful Quoted Bits

[Van Dyke \(1977\)](#) spends much of chapters four and five exploring the causes for the underrepresentation of certain groups in jury venires, and his analysis suggests that underrepresentation starts at the jury selection stage due to fewer non-white individuals registering to vote generally (page 89), and the process of applying to be excused from jury duty, in which economic hardship, which impacts disadvantaged economic groups to a greater extent (pages 113-120), is a common reason for excusal from jury duty.

Such explanations provide a plausible reason why black males would be most underrepresented in venires, and why the majority of the venire is white in this data despite the majority of defendants being black. Such issues with the jury selection process will not, and cannot, be solved by simply removed the peremptory challenge. They have much more to do with the relationship between certain groups and wider society, and so require more comprehensive and complex solutions.

Chapter 5

Summary

Summarize the presented work. Why is it useful to the research field or institute?

5.1 Future Work

Possible ways to extend the work.

Bibliography

- 42nd Parliament of Canada (2018a, March). Bill C-75: An Act to Amend the Criminal Code, Youth Criminal Justice Act and other Acts and to make consequential amendments to other Acts. <http://www.justice.gc.ca/eng/csj-sjc/pl/charter-charte/c75.html>.
- 42nd Parliament of Canada (2018b, November). Bill C75. LEGISinfo. <http://www.parl.ca/LegisInfo>.
- Attorney General's Office of the United Kingdom (2012, November). Jury vetting right of stand-by guidelines. <https://www.gov.uk/guidance/jury-vetting-right-of-stand-by-guidelines-2>.
- Baker, S. G. (1994). The multinomial-poisson transformation. *Journal of the Royal Statistical Society, Series D (The Statistician)* 43(4), 495–504.
- Baldus, D. C., G. Woodworth, D. Zuckerman, and N. A. Weiner (2001). The Use of Peremptory Challenges in Capital Murder Trials: A Legal and Empirical Analysis. *University of Pennsylvania Journal of Constitutional Law* 3(1).
- Brown, F. L., F. T. McGuire, and M. S. Winters (1978). The peremptory challenge as a manipulative device in criminal trials: Traditional use or abuse. *New England Law Review* 14, 192.
- Brown, R. B. (2000). Challenges for cause, stand-asides, and peremptory challenges in the nineteenth century. *Osgoode Hall Law Journal* 38(3), 453–494.
- Burgess, G. (1992). The divine right of kings reconsidered. *The English Historical Review* 107(425), 837–861.
- Cleveland, W. S. and R. McGill (1987). Graphical perception: The visual decoding of quantitative information on graphical displays of data. *Journal of the Royal Statistical Society. Series A (General)* 150(3), 192–229.
- Constitution of Canada (1982). Constitution of Canada. Accessed: <https://laws-lois.justice.gc.ca/eng/Const/index.html>.
- Constitution of the United States (1788). Constitution of the United States. Accessed: https://www.senate.gov/civics/constitution_item/constitution.htm.
- Ford, R. (2010). Modeling the effects of peremptory challenges on jury selection and jury verdicts. *George Mason Law Review* 17, 377.
- Forsyth, W. (1994). *History of Trial by Jury* (2 ed.). Lawbook Exchange.
- Friendly, M. (1994). Mosaic plots for multiway contingency tables. *Journal of the American Statistical Association* 89, 190–200.

- Government of Canada (1985). Criminal code section 634. <https://laws-lois.justice.gc.ca/eng/acts/C-46/section-634.html>.
- Government of Saskatchewan (1998). Jury act, 1998. Accessed: <http://www.qp.gov.sk.ca/documents/English/Statutes/Statutes/J4-2.pdf>.
- Grosso, C. M. and B. O'Brien (2012). A Stubborn Legacy: The Overwhelming Importance of Race in Jury Selection in 173 Post-Batson North Carolina Capital Trials. *Iowa Law Review* 97, 1531.
- Hans, V. P. and N. Vidmar (1986). *Judging the Jury* (1 ed.). Plenum Press.
- Harris, K. (2018, February). Liberals review jury selection process after Boushie case uproar. CBC News. <https://www.cbc.ca/news/politics/jury-selection-diversity-indigenous-1.4531792>.
- Hasan, N. R. (2018, April). Eliminating peremptory challenges makes trials less fair. The Star. <https://www.thestar.com/opinion/contributors/2018/04/10/eliminating-peremptory-challenges-make-trials-less-fair.html>.
- Hoffman, M. B. (1997). Peremptory Challenges Should Be Abolished: A Trial Judge's Perspective. *The University of Chicago Law Review* 64(3), 809.
- Iacobucci, F. (2013). First Nations Representation on Ontario Juries: Report of the Independent Review Conducted by the Honourable Frank Iacobucci. Ministry of the Attorney General. Accessed: https://www.attorneygeneral.jus.gov.on.ca/english/about/pubs/iacobucci/First_Nations_Representation.html.
- Kullback, S. and R. Leibler (1951). On information and sufficiency. *The Annals of Mathematical Statistics* 22(1), 79–86.
- Lang, J. B. (1996). On the comparison of multinomial and poisson log-linear models. *Journal of the Royal Statistical Society, Series B (Methodological)* 58(1), 253–266.
- MacLean, C. (2018, February). Gerald Stanley acquittal renews calls for justice reform 27 years after Manitoba inquiry. CBC News. <https://www.cbc.ca/news/canada/manitoba/aboriginal-justice-inquiry-colten-boushie-gerald-stanley-jury-1.4532394>.
- Macnab, A. (2018, February). Stanley acquittal should not lead to scrapping peremptory challenges, say criminal lawyers. Canadian Lawyer. <https://www.canadianlawyermag.com/legalfeeds/author/aidan-macnab/stanley-acquittal-should-not-lead-to-scrapping-peremptory-challenges-say-criminal-lawyers-15332/>.
- Ministry of the Attorney General of Ontario (2018). The annual jury selection process. Queen's Printer for Ontario. Accessed: https://www.attorneygeneral.jus.gov.on.ca/english/courts/jury/jury_selection_process.php.
- Mirriam-Webster (2019a). Mirriam-Webster Dictionary Online. Accessed: <https://www.merriam-webster.com/dictionary/venire>.
- Mirriam-Webster (2019b). Mirriam-Webster Dictionary Online. Accessed: <https://www.merriam-webster.com/dictionary/voir%20dire>.

- Morehead, J. W. (1994). When a peremptory challenge is no longer peremptory: Batson's unfortunate failure to eradicate invidious discrimination from jury selection. *DePaul Law Review* 43, 625.
- Nisbett, R. E. and Z. Kunda (1985). Perception of social distributions. *Journal of Personality and Social Psychology* 48(2), 297–311.
- North Carolina Sentencing and Policy Advisory Commission (2017). *Classification of a Sample of Offenses*. North Carolina Judicial Branch. Accessed: <https://www.nccourts.gov/assets/documents/publications/Sample-list-2017.pdf?MAZluXuS0FWod4nFt7zXecJ8Ifu0qEZx>.
- Page, A. (2005). Batson's blind spot: Unconscious stereotyping and the peremptory challenge. *Boston University Law Review* 85, 155.
- Petersen, C. (1993). Insitutionalized racism: The need for reform of the criminal jury selection process. *McGill Law Journal* 38(1).
- Quenneville, G. (2018, February). What happened on Gerald Stanley's farm the day Colten Boushie was shot, as told by witnesses. CBC News. <https://www.cbc.ca/news/canada/saskatoon/what-happened-stanley-farm-boushie-shot-witnesses-colten-gerald-1.4520214>.
- Quenneville, G. and J. Warick (2018, February). Shouts of 'murderer' in courtroom after Gerald Stanley acquitted in Colten Boushie shooting. CBC News. <https://www.cbc.ca/news/canada/saskatoon/gerald-stanley-colten-boushie-verdict-1.4526313>.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.
- Revesz, J. (2016). Ideological imbalance and the peremptory challenge. *Yale Law Journal* 125.
- Richardson Oakes, A. and H. Davies (2016). Justice must be seen to be done: a contextual reappraisal. *Adelaide Law Review* 37(2), 461–494.
- Roach, K. (2018, April). Ending peremptory challenges in jury selection is a good first step. The Ottawa Citizen. <https://ottawacitizen.com/opinion/columnists/roach-ending-peremptory-challenges-in-jury-selection-is-a-good-first-step>.
- Sheppard, R. (2018). Patriation of the constitution. The Canadian Encyclopedia: Historical Canada. Accessed: <https://www.thecanadianencyclopedia.ca/en/article/patriation-of-the-constitution>.
- Sison, C. P. and J. Glaz (1995). Simultaneous confidence intervals and sample size determination for multinomial proportions. *Journal of the American Statistical Association* 90(429), 366–369.
- Statistics Canada (2018, November). Table 35-10-0061-01: Crime severity index and weighted clearance rates, police services in Saskatchewan.
- Supreme Court of Canada (1991). *R. v. sherratt*. Supreme Court Judgments. SCC Case Number: 21501; Accessed: <https://scc-csc.lexum.com/scc-csc/scc-csc/en/item/734/index.do?q=21501>.

- Supreme Court of the United States (1965). *Swain v. Alabama*. Accessed: <https://supreme.justia.com/cases/federal/us/380/202/>.
- Supreme Court of the United States (1986). *Batson v. Kentucky*. Accessed: <https://www.law.cornell.edu/supremecourt/text/476/79>.
- Supreme Court of the United States (1993). *J.E.B. v. Alabama*. Accessed: <https://supreme.justia.com/cases/federal/us/511/127/>.
- Tufte, E. R. (2001). *The Visual Display of Quantitative Information* (2 ed.). Graphics Press.
- Van Dyke, J. M. (1977). *Jury Selection Procedures: Our Uncertain Commitment to Representative Panels* (1 ed.). Ballinger Publishing.
- Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (Fourth ed.). New York: Springer. ISBN 0-387-95457-0.
- von Moschzisker, R. (1921). The historic origin of trial by jury. *University of Pennsylvania Law Review* 70(1).
- Wegman, E. J. (1990). Hyperdimensional analysis using parallel coordinates. *Journal of the American Statistical Association* 85(411), 664–675.
- Weisstein, E. W. (2018). Chebyshev inequality. MathWorld - A Wolfram Web Resource. Accessed: <http://mathworld.wolfram.com/ChebyshevInequality.html>.
- Wickham, H. and J. Bryan (2018). *readxl: Read Excel Files*. R package version 1.1.0.
- Woolley, A. (2018). An Ethical Jury? Reflections on the Acquittal of Gerald Stanley for the Murder/Manslaughter of Colten Boushie. *Slaw: Canada's online legal magazine*. Accessed: <http://www.slaw.ca/2018/02/20/an-ethical-jury-reflections-on-the-acquittal-of-gerald-stanley-for-the-murder-manslaughter-of-colten-boushie/>.
- Wright, R. F., K. Chavis, and G. S. Parks (2018, October). The Jury Sunshine Project: Jury Selection Data as a Political Issue. *University of Illinois Law Review* 2018(4), 1407.
- Zinchuk, B. (2018, March). Both sides wrong about Stanley trial. Prince George Citizen. <https://www.princegeorgecitizen.com/opinion/editorial/both-sides-wrong-about-stanley-trial-1.23199321>.

Appendix A

Developing an Effective Visualization of Conditional Probability

One deficiency of the results of the previous investigations was a failure to generate compelling and effective visualizations of the trends of peremptory challenges for different racial groups. While such visualizations are not necessarily critical to analysis, they can often be incredibly useful to not only communicate data, but to motivate further investigations and models in a way which is clearer and more intuitive than a simple table of values.

The first attempt at such a visualization was the mosaic plot (as discussed by [Friendly \(1994\)](#)) using the `mosaicplot` function in the `graphics` package in R ([R Core Team \(2018\)](#)). Figure [A.1](#) displays this first approach with disposition related to the simplified races of both the defendant and the venire member.

This visualization suffers from a number of limitations, some of which are obvious, and others of which are best explained by the hierarchy of accuracy of visual perception provided in [Cleveland and McGill \(1987\)](#). The obvious limitations are the lack of ability to perceive the differences for the smallest groups, which are compressed enough that their error is nearly imperceptible. Additionally, the ordering of the axes is incredibly important in how the different areas appear visually, and comparing the different areas is unclear if any specific comparisons are to be made.

This may be somewhat unsurprising. [Cleveland and McGill \(1987\)](#), in their ranking of visual displays by accuracy of perception place area low in the hierarchy, below angles, lengths, and positions along common scales. In *The Visual Display of Quantitative Information*, [Tufte](#) gives two more sources of possible criticism of the mosaic plot as displayed in Figure [A.1](#): the concept of data-ink and the dimensionality of visualization.

Of the mosaic plot, one may ask how much of the “ink”, or structure, on the page is necessary to communicate the information present. If one has a desire to “above all else show the data” as Tufte does, then these large shaded rectangles, which are likely not perceived accurately according to [Cleveland and McGill](#), seem unnecessary compared to a simpler visualization. This is the concept of “data-ink,” to reduce the complexity of the structures and chart used to display the data.

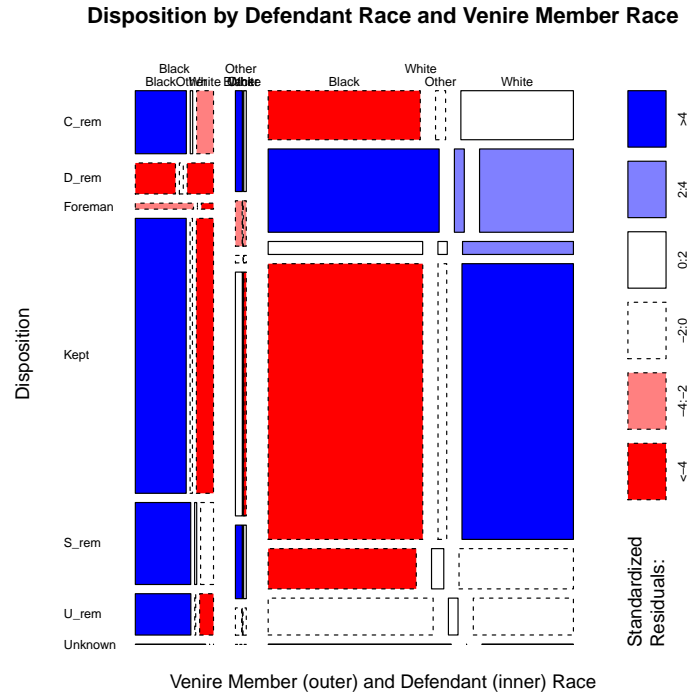


Figure A.1: A mosaic plot of the simplified defendant and venire member race and their relation to the disposition of the venire member.

Hand-in-hand with this concept for this plot is [Tufte's](#) rule that the dimensionality of the visualization should not be larger than the data. In the case of the mosaic plot this is not strictly violated, as the marginal lengths used to create the areas reflect a measurement of the data. Nonetheless, the areas of each rectangle correspond to a simple count in a contingency table, and perhaps an area is not the best way to represent such a singular value.

Motivated by these concepts, parallel coordinates (as in [Wegman \(1990\)](#)) were used to visualize the data next, as can be seen in [Figure A.2](#). This attempted visualization is arguably more difficult to interpret than the mosaic plot. It is cluttered by the parallel coordinate lines, the bars emanating from each point obscure the fact that the end point of the bar is the only feature of interest, and the meaning of the black reference line is entirely unclear without extensive explanation. Finally, by viewing the distribution of each disposition, the wrong conditional density is being examined, $P(\text{Race}, \text{Race}_{\text{Defendant}} | \text{Disposition})$. Multiple edits and re-conceptualizations of the concept eventually resulted in [Figure ??](#), which will be called the “mobile plot” due to its passing resemblance to the mobiles hung above babies’ cribs.

An example of this plot can be seen in [4.1](#). Note that this plot is less cluttered than either the mosaic plot or the first parallel coordinate plot, despite displaying more information. It is also more efficient with data-ink, avoids displaying data with higher dimensions than the data itself, and uses redundant encoding of information in visual cues which are high in the hierarchy presented by [Cleveland and McGill \(1987\)](#).

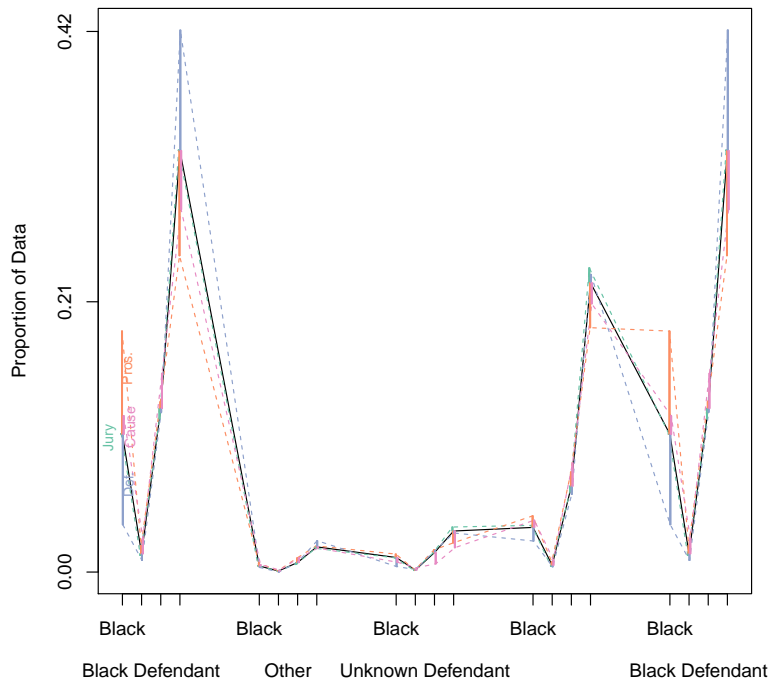


Figure A.2: The first attempt at a parallel coordinate plot attempted. Note that the cramped display and unclear definition of the axis make interpretation even less intuitive than the mosaic plot, suggesting that this first attempt was a decided failure.

An explanation of the features and encoding used in the mobile plot is presented in [A.1](#).

A.1 The Mobile Plot

The mobile plot consists of multiple grouped vertical lines anchored at one end to horizontal black lines, and at the other to points. Information is encoded using length, colour, and position relative to a common scale. The vertical axis is meant to show the value of a continuous variable, while the horizontal axis shows the value of a, possibly hierarchical, categorical variable. It can be used to display the relationship between three categorical variables and a continuous variable in a meaningful two-dimensional plot.

To show the grouping of categories on the horizontal axis, position is used. Those categorical levels which are grouped by some separate categorical variable are placed closer to each other than those which are not in the same group. Each categorical variable combination corresponds to a single horizontal black line, the length of which is proportional to the count of the associated combination in the data being plotted. The vertical position of this line corresponds to the value of the continuous variable expected for that particular combination.

Each of the vertical lines which extend from this horizontal line corresponds to a particular value of a third categorical variable, coloured to show the specific level across the

different horizontal lines. The end points of these lines represent the observed value of the continuous variable for the three way combination of categorical variables represented by the vertical and horizontal line combination. The lengths of these lines correspond to the deviation of the observation from the expectation. If a different expectation is expected for the different values of the third categorical variable, the horizontal lines can be split evenly and placed vertically at this expectation, to the detriment of grouping clarity.

In the case that such a split is not used and the continuous variable is the probability of a particular value of the third categorical variable given the first two, the plot serves as a visual test of a very specific hypothesis: that of a uniform distribution of the third categorical variable with respect to the two variables represented horizontally. Such a plot is powerful because it allows for the simple detection of main effects and interaction effects over the three categorical variables against this hypothesis.

Appendix B

Complementary information

Additional material. For example long mathematical derivations could be given in the appendix. Or you could include part of your code that is needed in printed form. You can add several Appendices to your thesis (as you can include several chapters in the main part of your work).

B.1 Jury Sunshine Irregularities

Table B.1: Jury sunshine data irregularities noted in data flattening

Charges without trial (ACISID)	08CRS50940, 09CRS1106, 10CRS051975, 10CRS51388, 11CRS051642, 11CRS1745, 11CRS51895, 08CRS50113	08CRS52888, 09CRS50752, 10CRS1215, 10CRS51610, 11CRS051795, 11CRS1783, 11CRS52470,	09CRS000305, 10CR52031, 10CRS397, 10CRS52410, 11CRS1577, 11CRS51204, 08CRS54836,
Prosecutors without trials (IDs)	1-000, 11B-000, 12-000, 14-000, 15B-000, 16A-000, 16B-000, 17A-000, 17B-000, 19A-000, 19B-000, 20A-000, 20B-000, 21-000, 22A-000, 22B-000, 24-000, 25-000, 27A-000, 27B-000, 28-000, 29A-000, 29B-000, 30-000, 6-000, 9-000		
Trial missing charge (ID)	710-01		

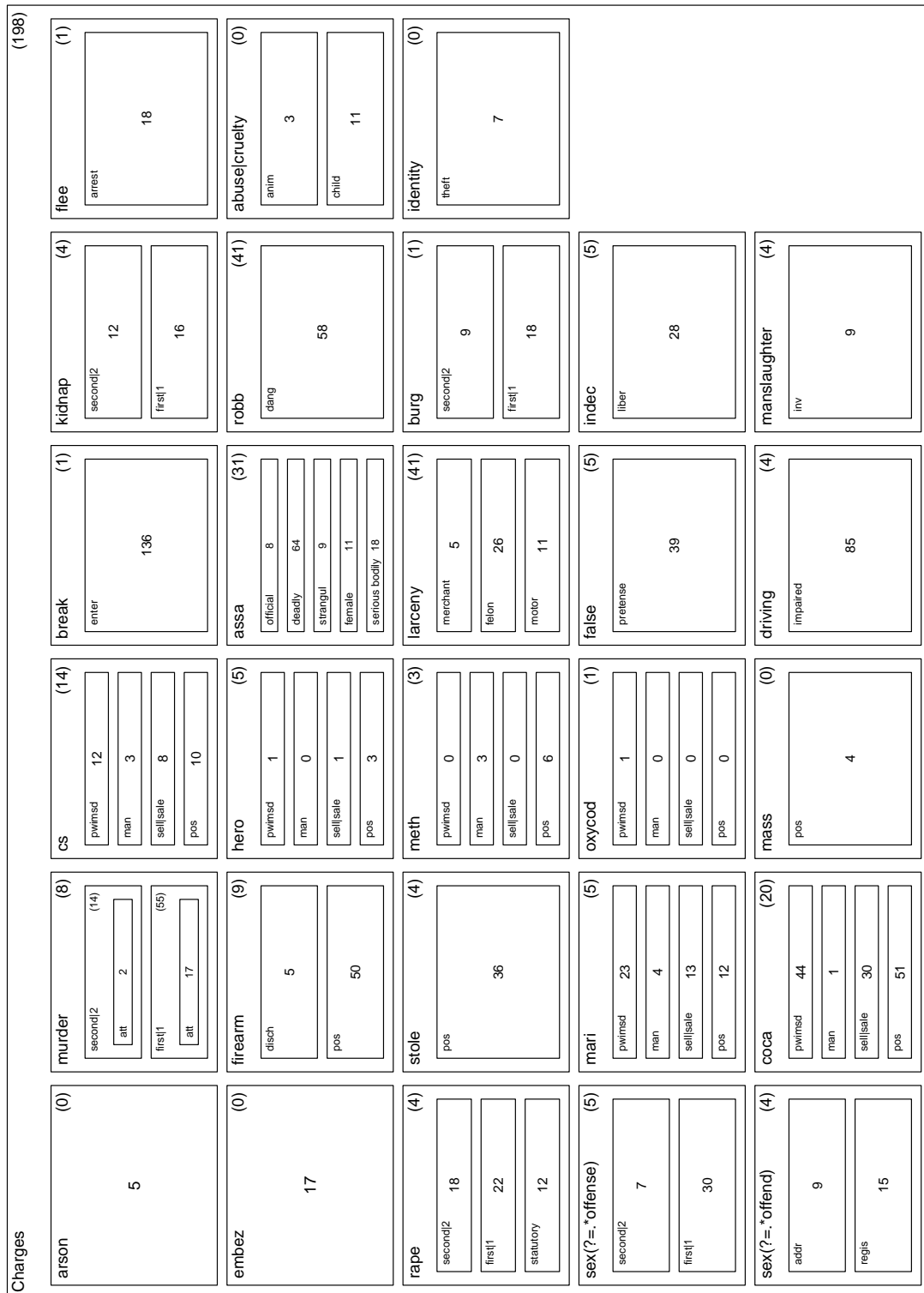


Figure B.1: The regular expression charge tree arranged by hierarchy with counts provided. The counts in brackets indicate the counts of charges which could not be classified to a lower level of the hierarchy

B.2 Jury Sunshine Charge Classification

B.3 Using Sweave to include R code (and more) in your report

The easiest (and most elegant) way to include R code and its output (and have all your figures up to date with your report) is to use Sweave. You can find an introduction Sweave in `/u/sfs/StatSoftDoc/Sweave/Sweave-tutorial.pdf`.

Appendix C

Mathematical Results

C.1 Conditional Distribution of a Poisson Expectation Given Marginal Counts

For simplicity, consider a Poisson random variable Y with a rate dependent only on one discrete random variable: $X \in \{1, \dots, m\}$. Let the count of values observed for $X = i$ be Y_i and denote $E[Y_i|X = i] = \lambda_i$. Additionally denote the sum of all counts as $N = \sum_{i=1}^m Y_i$. The feature of interest is then the distribution of $Y_1, \dots, Y_m|N = n$. Or by the definition of the conditional distribution:

$$P(Y_1 = y_1, \dots, Y_m = y_m|N = n) = \frac{P(Y_1 = y_1, \dots, Y_m = y_m, \sum_{i=1}^m Y_i = n)}{P(\sum_{i=1}^m Y_i = n)} \quad (\text{C.1.0.1})$$

Clearly this density is zero if $\sum_{i=1}^m y_i \neq n$, but consider its value with for $\sum_{i=1}^m y_i = n$. Start with the distribution of $N = \sum_{i=1}^m Y_i$. Note that for $A \sim \text{Pois}(\lambda_A)$ and $B \sim \text{Pois}(\lambda_B)$, where A and B are independent, the distribution of $A + B$ can be derived quite easily using the characteristic function $\varphi_{A+B}(t)$:

$$\varphi_{A+B}(t) = E[e^{it(A+B)}] = E[e^{itA}]E[e^{itB}] = e^{(\lambda_A + \lambda_B)(e^{it} - 1)}$$

This is the characteristic function of a $\text{Pois}(\lambda_A + \lambda_B)$ variable, and so the sum of two Poisson random variables is a Poisson random variable with a rate corresponding to the sum of the two variables. Iterating this, then, one obtains $N \sim \text{Pois}(\sum_{i=1}^m \mu_i)$, and so the denominator of C.1.0.1 is:

$$P\left(\sum_{i=1}^m Y_i = n\right) = \frac{e^{-\sum_{i=1}^m \mu_i} (\sum_{i=1}^m \mu_i)^n}{n!}$$

Additionally, recognizing that the Y_{ij} are independent, and considering only the case where $\sum_{i=1}^m y_i = n$, as the density is zero otherwise, this can be further simplified, as this joint density can be split into a product of marginal densities:

$$P\left(Y_1 = y_1, \dots, Y_m = y_m, \sum_{i=1}^m y_i = n\right) = P(Y_1 = y_1)P(Y_2 = y_2) \dots P(Y_m = y_m)$$

Now each independent marginal is Poisson distributed, so the product of all of these marginals is:

$$P(Y_1 = y_1)P(Y_2 = y_2) \dots P(Y_m = y_m) = \frac{e^{\sum_{i=1}^m \mu_i} \mu_1^{y_1} \mu_2^{y_2} \dots \mu_m^{y_m}}{y_1! y_2! \dots y_m!}$$

And so C.1.0.1 simplifies to

$$\begin{aligned} & \frac{e^{\sum_{i=1}^m \mu_i} \mu_1^{y_1} \mu_2^{y_2} \dots \mu_m^{y_m}}{y_1! y_2! \dots y_m!} \cdot \frac{n!}{e^{\sum_{i=1}^m \mu_i} \left(\sum_{i=1}^m \mu_i\right)^n} \\ &= \frac{m!}{y_1! y_2! \dots y_m!} \left(\frac{\mu_1}{\sum_{i=1}^m \mu_i}\right)^{y_1} \left(\frac{\mu_2}{\sum_{i=1}^m \mu_i}\right)^{y_2} \dots \left(\frac{\mu_m}{\sum_{i=1}^m \mu_i}\right)^{y_m} \end{aligned} \quad (\text{C.1.0.2})$$

C.1.0.2 is recognizably the multinomial distribution, where the probability of a particular class i is given by the ratio of μ_i to the sum over all μ_j . ■

Appendix D

Code

D.1 Data Processing Code

Below is all of the code used to load and process the data, including both function definitions and the script which details the order and method of their calling. A more convenient way of accessing the code is through the author's GitHub (https://github.com/Salahub/peremptory_challenges), where all of the code and some basic data files are posted.

```
1 #####
2
3 ## THESIS DATA PROCESSING SCRIPT
4 ## Christopher Salahub
5 ## Sept 26, 2018
6
7 #####
8
9 ## PACKAGES #####
10 library(readxl)
11 library(tm)
12 library(stringr)
13 library(grid)
14
15
16 ## CONSTANTS #####
17
18 ## start by defining file locations
19 ThesisDir ← "c:/Users/Chris/Documents/ETH Zurich/Thesis/Data"
20 SunshineFile ← paste0(ThesisDir, "/JurySunshineExcel.xlsx")
21 SunshineSheets ← excel_sheets(SunshineFile)
22
23 NorthCarFile ← paste0(ThesisDir,
24                       "/Jury Study Data and Materials/NC Jury Selection Study
25                        Database6 Dec 2011.csv")
26
27 PhillyFile ← paste0(ThesisDir,
28                     "/Voir Dire Data & Codebook/capital_venires.csv")
29
30 ## next the factor level codes as given in the codebook and regularized here
31 ## regularization: - political affiliation "N" replaced with "I" for all entries
32 LevRace ← sort(c("A", "B", "H", "N", "O", "U", "W"))
33 LevGen ← sort(c("F", "M", "U"))
34 LevPol ← sort(c("D", "L", "R", "I", "U"))
35
36 ## create a charge tree with regex nodes to identify and clean charge text
37 chargeTree ← list("rape" = list("statutory", "first|1", "second|2"), "sex(?=.*
38 offense)" = list("first|1", "second|2"),
```

```

37     "sex(?.*offend)" = list("regis", "addr"), "murder" = list("
38         first|1" = list("att"), "second|2" = list("att")),
39     "arson", "firearm" = list("pos", "disch"), "stole" = list("pos
40     "),
41     "mari" = list("pos", "sell|sale", "man", "pwimsd"), "coca" =
42     list("pos", "sell|sale", "man", "pwimsd"),
43     "cs" = list("pos", "sell|sale", "man", "pwimsd"), "hero" =
44     list("pos", "sell|sale", "man", "pwimsd"),
45     "meth" = list("pos", "sell|sale", "man", "pwimsd"),
46     "oxycod" = list("pos", "sell|sale", "man", "pwimsd"), "mass" =
47     list("pos", "break" = list("enter")),
48     "assa" = list("serious bodily", "female", "strangul", "deadly"
49     , "official"),
50     "larceny" = list("motor", "felon", "merchant"), "false" = list
51     ("pretense"),
52     "driving" = list("impaired"), "kidnap" = list("first|1", "
53     second|2"),
54     "robb" = list("dang"), "burg" = list("first|1", "second|2"), "
55     indec" = list("liber"),
56     "embez", "manslaughter" = list("inv"), "flee" = list("arrest")
57     ,
58     "abuse|cruelty" = list("child", "anim"), "identity" = list("
59     theft"))
60
61 ## create a list of variables which can sensibly be summarized by trial
62 TrialVars <- c("TrialNumberID", "DateOutcome", "JudgeID", "DefAttyType", "
63     VictimName",
64     "VictimRace", "VictimGender", "CrimeLocation", "PropertyType",
65     "ZipCode.Trials", "StateTotalRemoved", "DefenseTotalRemoved",
66     "CourtTotalRemoved", "JDistrict", "JName", "JRace", "JGender",
67     "JPoliticalAff", "JVoterRegYr", "JYrApptd", "JResCity", "JResZip",
68     "ChargeTxt", "Outcome", "Sentence.FullSunshine", "DefendantID.
69     FullSunshine",
70     "DefendantID.DefendantToTrial", "DefRace", "DefGender", "DefDOB",
71     "DefAttyID",
72     "DefAttyName", "DCRace", "DCGender", "DCPoliticalAff", "
73     DCYrRegVote",
74     "DCYrLicensed", "DCResideCity", "DCResideZip", "ProsecutorID", "
75     ProsName",
76     "ProsRace", "ProsGender", "ProsPoliticalAff", "PYrRegVote", "
77     PYrLicensed",
78     "PResideCity", "PResideZip", "Guilty", "CrimeType", "DefWhiteBlack
79     ")
80
81 ## FUNCTIONS #####
82
83 ## Loading and cleaning #####
84 ## create a descriptive merge function for cleaning (essentially a 'merge'
85     wrapper)
86 CleaningMerge <- function(x, y, ...) {
87     ## start by creating the merge
88     ## first match arguments
89     MatchCall <- match.call(merge)
90     MatchCall[[1]] <- quote(merge)
91     ## get input names and ensure proper name structure
92     xname <- MatchCall$x
93     if (!is.symbol(xname)) xname <- as.symbol(paste0(xname[[2]], xname[[3]]))
94     yname <- MatchCall$y
95     if (!is.symbol(yname)) yname <- as.symbol(paste0(yname[[2]], yname[[3]]))
96     ## use this to extract suffixes and fix MatchCall
97     MatchCall$suffixes <- paste0(".", c(xname, yname))
98     MatchCall$x <- xname
99     MatchCall$y <- yname
100    ## specify that the match should be an outer join
101    MatchCall$all <- TRUE
102    ## and use this to make a clean local assignment to modify
103    assign(as.character(xname), cbind(x, Diag.x = 1), envir = environment())
104    assign(as.character(yname), cbind(y, Diag.y = 1), envir = environment())
105    ## now evaluate the call

```

```

87 Merged <- eval(MatchCall, envir = environment())
88 ## next perform some checks
89 xExpInds <- is.na(Merged$Diag.x)
90 yExpInds <- is.na(Merged$Diag.y)
91 ## remove the diagnostic columns
92 Merged$Diag.x <- NULL; Merged$Diag.y <- NULL
93 ## summarize the diagnostic checks
94 X_nexp <- sum(xExpInds)
95 Y_nexp <- sum(yExpInds)
96 X_missing <- Merged[xExpInds,]
97 Y_missing <- Merged[yExpInds,]
98 ## print the diagnostics
99 cat("Joined ", paste(xname, yname, sep = " and "), " with ",
100     X_nexp, " and ", Y_nexp, " failed matches respectively \n", sep = "")
101 ## return the results, preferentially keeping the data which is present in x
102     but missing from y
103 if (X_nexp == 0 & Y_nexp == 0) {
104   Merged
105 } else list(Merge = Merged[!xExpInds,], Xfails = X_missing, Yfails = Y_
106     missing)
107 }
108 ## a function to identify and perform swaps with user input
109 SimpleSwapper <- function(data, CorrectLevs, auto = FALSE) {
110   ## first match the data to the columns of interest
111   colInds <- match(names(CorrectLevs), names(data))
112   ## extract the levels of the columns of interest to check if there are any
113     potential swaps
114   swapCheck <- all(sapply(1:length(colInds),
115     function(ind) identical(sort(levels(as.factor(data[,
116       colInds[ind]]))),
117       sort(CorrectLevs[[ind]])))
118   ## if no swaps are present end this check
119   if (swapCheck) {
120     cat("No errors found, exiting.")
121     return(data)
122   }
123   ## if errors are found, further investigate them
124   ## identify potential rows
125   ## first those which have elements out of place
126   SwapPoss <- sapply(1:length(colInds),
127     function(ind) !(data[,colInds[ind]] %in% CorrectLevs[[ind]]))
128   ## now rows containing unknown entries
129   Unknown <- sapply(1:length(colInds),
130     function(ind) data[,colInds[ind]] == "U")
131   ## identify potential swaps by row
132   Swaps <- apply(SwapPoss, 1, function(row) sum(row) > 1)
133   ## identify the potential errors
134   PotErr <- apply(SwapPoss, 1, function(row) sum(row) == 1)
135   ## use the unknowns to account for some errors
136   UnkInd <- apply(Unknown, 1, any)
137   FalErr <- PotErr & UnkInd
138   ## identify the indices to investigate
139   SwapInds <- which(Swaps|FalErr)
140   ErrInds <- which(PotErr & !UnkInd)
141   ## communicate to the user and ask for input
142   cat("There are ", sum(Swaps|FalErr), " swaps to check\n", sep = "")
143   cat("Additionally, it seems there are ", sum(PotErr & !UnkInd), " errors in
144     entries\n", sep = "")
145   ## unless automated
146   if (auto) ErrorReturn <- TRUE else ErrorReturn <- as.logical(readline("Return
147     the errors? (T/F): "))
148   ## now, if there are possible swaps investigate them
149   if (sum(Swaps|FalErr) != 0) {
150     ## create a temporary storage structure
151     tempRows <- data[SwapInds, colInds]
152     tempRows <- as.data.frame(lapply(tempRows, function(var) levels(var)[as.
153       numeric(var)]),
154       stringsAsFactors = FALSE)

```

```

150     ## loop through and populate this
151     for (ii in 1:nrow(tempRows)) {
152         ## inspect the row
153         print(tempRows[ii,])
154         ## suggest corrections, first generate matches
155         candComb ← lapply(tempRows[ii,],
156                             function(el) which(sapply(CorrectLevs,
157                                                         function(levs) el %in%
158                                                             levs)))
159
160         reps ← unlist(lapply(candComb, length))
161         ## now generate all swap combinations
162         candComb[[1]] ← rep(candComb[[1]], each = max(reps[-1]))
163         candComb ← as.data.frame(candComb, row.names = NULL)
164         ## identify rows which contain all indices, in other words those
165         ## valid as swaps
166         compRows ← apply(candComb, 1, function(row) all(1:length(CorrectLevs)
167                                                         %in% row))
168         goodComb ← candComb[compRows,]
169         ## clean them up and print them
170         colnames(goodComb) ← NULL
171         rownames(goodComb) ← NULL
172         cat("Potential combinations:\n")
173         print(t(apply(goodComb, 1, order)))
174         ## take user input or automatically determine value
175         if (auto) {
176             if (!any(compRows)) acceptedComb ← 0 else acceptedComb ← 1
177         } else acceptedComb ← as.numeric(readline("Enter a combination choice
178             (0 for error, <enter> to accept first): "))
179         ## handle special cases, 0 if a true error has been identified
180         if (identical(acceptedComb, 0)) { ## 0 if a true error has been
181             identified
182             ErrInds ← c(ErrInds, SwapInds[ii])
183             cat("True error identified, adding ", SwapInds[ii], " to error
184                 list\n", sep = "")
185         } else { ## the case where a swap has been correctly identified and
186             selected, or enter has been pressed
187             ## if enter has been pressed accept the first row
188             if (is.na(acceptedComb)) acceptedComb ← 1
189             ## print recombined row
190             newRows ← tempRows[ii, order(as.matrix(goodComb[acceptedComb,]))]
191             colnames(newRows) ← NULL
192             rownames(newRows) ← NULL
193             cat("Corrected row:")
194             print(newRows)
195             cat("-----\n")
196             ## correct entry
197             tempRows[ii,] ← newRows
198         }
199     }
200     ## fill the data
201     ## first prevent factor level errors
202     data[, colInds] ← lapply(colInds, function(ind) levels(data[, ind])[as.
203         numeric(data[, ind])])
204     ## now swap the data
205     data[SwapInds, colInds] ← lapply(1:length(colInds), function(ind) tempRows
206         [, ind])
207     ## reconvert back to factors
208     data[, colInds] ← lapply(colInds, function(ind) as.factor(data[, ind]))
209 }
210
211 ## in either case return the data and errors as specified
212 if (ErrorReturn) {
213     return(list(Data = data, Errors = ErrInds))
214 } else {
215     return(data)
216 }
217
218 ## now create a function to address the errors possibly identified in the above
219 function automatically
220 SwapErrorFix ← function(errorData, CorrectLevs) {

```



```

210  ## check if we are in the case without errors
211  if (!identical(names(errorData), c("Data", "Errors"))) {
212    cat("No errors\n")
213    return(errorData)
214  } else {
215    ## extract the data and data in error
216    fulldata ← errorData$Data
217    ## get the relevant columns
218    colInds ← match(names(CorrectLevs), names(fulldata))
219    ## go through the specified variables and remove errors
220    fixed ← lapply(1:length(colInds),
221      function(ind) {
222        var ← fulldata[, colInds[ind]]
223        var ← levels(var)[as.numeric(var)]
224        inds ← !(var %in% CorrectLevs[[ind]])
225        cat(names(CorrectLevs)[ind], ": ", sum(inds),
226          " errors\n", sep = "")
227        var[inds] ← "U"
228        as.factor(var)
229      })
230    ## insert these fixed values
231    fulldata[, colInds] ← fixed
232    ## return this
233    fulldata
234  }
235 }
236
237 ## write a wrapper to perform this swapping and error correction in one call
238 SwapandError ← function(data, CorrectLevs) {
239   swapped ← SimpleSwapper(data = data, CorrectLevs = CorrectLevs, auto = TRUE)
240   fixed ← SwapErrorFix(errorData = swapped, CorrectLevs = CorrectLevs)
241   fixed
242 }
243
244 ## Variable Synthesis #####
245 ## Kullback-Leibler divergence function
246 kldiv ← function(samp, dist) {
247   ## convert to matrices
248   mat1 ← as.matrix(samp)
249   mat2 ← as.matrix(dist)
250   ## make into proper distributions
251   mat1 ← mat1/rowSums(mat1)
252   mat2 ← mat2/rowSums(mat2)
253   ## take the log ratio
254   logratio ← log(mat1/mat2)
255   ## multiply by correct matrix
256   vals ← mat1*logratio
257   ## take the row sums
258   rowSums(vals, na.rm = TRUE)
259 }
260
261 ## make a text-mining regularization function
262 StringReg ← function(strs) {
263   ## first set everything to lowercase
264   strs ← tolower(strs)
265   ## replace specific patterns (noticed during early tests)
266   strs ← str_replace_all(strs, "b/e|break/enter|b&e|break or enter|b or e|b &/
    or e|b & e", "breaking and entering")
267   strs ← str_replace_all(strs, "controlled substance", "cs")
268   strs ← str_replace_all(strs, "dwi", "driving while impaired")
269   strs ← str_replace_all(strs, "rwdw", "robbery with a deadly weapon")
270   strs ← str_replace_all(strs, "pwisd|pwmsd|pwmsd|pwitd|pwid|pwmisd|pwosd", "
    pwimsd")
271   strs ← str_replace_all(strs, "robbery|rob ", "robbery")
272   strs ← str_replace_all(strs, "bulgary", "burglary")
273   strs ← str_replace_all(strs, "awdw", "assault with a deadly weapon")
274   strs ← str_replace_all(strs, "(?<=[\\sa-z])[0-9]{2,}", "")
275   strs ← str_replace_all(strs, "att ", "attempted ")
276   strs ← str_replace_all(strs, "assult", "assault")
277   strs ← str_replace_all(strs, "marj", "marijuana")

```

```

278   ## replace punctuation
279   strs ← gsub("[^[:alnum:][:space:]]'", "", strs)
280   ## return these
281   strs
282 }
283
284 ## create a function to process such a tree structure given a list of strings
285 stringTree ← function(strs, regexTree, inds = 1:length(strs), includeOther = TRUE)
286 {
287   ## identify the sublists, and divide the data
288   sublists ← sapply(regexTree, is.list)
289   ## iterate over unnamed items (leaf nodes)
290   listdiv ← lapply(regexTree[!sublists], function(el) inds[grepl(el, strs, perl
291     = TRUE)])
292   names(listdiv) ← unlist(regexTree[!sublists])
293   ## check if there are any sublists
294   if (!any(sublists)) {
295     if (includeOther) listdiv ← c(listdiv, other = list(inds[!(inds %in%
296       unlist(listdiv))]))
297     ## in the case of none, treat the object as a list to iterate through
298     listdiv
299   } else {
300     ## otherwise recurse over the branches
301     finlist ← c(listdiv, lapply(names(regexTree)[sublists],
302       function(name) stringTree(strs[grepl(name,
303         strs, perl = TRUE)],
304         regexTree[[name]],
305         inds[grepl(name,
306           strs, perl =
307             TRUE)],
308         includeOther)))
309     names(finlist)[(length(listdiv) + 1):length(finlist)] ← names(regexTree)[
310       sublists]
311     c(finlist, other = list(inds[!(inds %in% unlist(finlist))]))
312   }
313 }
314
315 ## create a tree depth helper function
316 maxdepth ← function(tree, counter = 1) {
317   max(sapply(tree, function(br) if (!is.list(br)) counter else maxdepth(br,
318     counter + 1)))
319 }
320
321 ## create a function to aggregate a tree as specified above at the desired depth
322 treeAgg ← function(tree, level = 1) {
323   ## first check the max depth of the tree
324   treedepth ← maxdepth(tree)
325   ## compare this to requested aggregation level
326   stopifnot(level <= treedepth)
327   ## aggregate at desired level with a helper function
328   agg ← function(tr, depth = 1) {
329     if (depth == level) lapply(tr, function(el) setNames(unlist(el), NULL))
330     else lapply(tr, function(br) agg(dr, depth + 1))
331   }
332   agg(tree)
333 }
334
335 ## create a crime class aggregation function
336 CrimeClassify ← function(tree, regChar) {
337   crimes ← list()
338   crimes$Sex ← unique(c(unlist(tree[c("rape", "sex(=?.*offense)", "sex(=?.*
339     offend)", "indec"])),
340     tree$other[grepl("sex", regChar[tree$other])]))
341   crimes$Theft ← unique(unlist(tree[c("stole", "embez", "break", "larceny", "
342     robb", "burg", "identity"])))
343   crimes$Murder ← unique(unlist(tree[c("murder", "manslaughter"])))
344   crimes$Drug ← unique(c(unlist(tree[c("mari", "coca", "cs", "hero", "meth", "
345     oxycod"])),
346     tree$other[grepl("para|drug|substance|pwmsd",
347       regChar[tree$other])]))

```

```

335     crimes$Violent ← unique(unlist(tree[c("arson", "assa", "abuse|cruelty"))))
336     crimes$Driving ← unique(c(unlist(tree[c("driving")])),
337                               tree$other[grepl("hit(?:.*run)|speeding", regChar[
338                                     tree$other], perl = TRUE)))]))
339 }
340
341 ## in order to make the process of pre-processing the data and adding desired
342 ## flexible function and add operations as desired
343 SynCols ← function(data) {
344     ## too busy, synthesize some variables to clearly indicate the results of
345     ## defense and prosecution selection
346     data$VisibleMinor ← data$Race != "White"
347     data$PerempStruck ← grepl("S_rem|D_rem", data$Disposition)
348     data$DefStruck ← data$Disposition == "D_rem"
349     data$ProStruck ← data$Disposition == "S_rem"
350     data$CauseRemoved ← data$Disposition == "C_rem"
351     ## lets look at which race struck each juror
352     data$StruckBy ← as.factor(sapply(1:nrow(data),
353                                     function(ind) {
354                                         dis ← as.character(data$
355                                             Disposition[ind])
356                                         if (dis == "S_rem") {
357                                             as.character(data$ProsRace
358                                                 [ind])
359                                         } else if (dis == "D_rem") {
360                                             as.character(data$DCRace[
361                                                 ind])
362                                         } else "Not Struck"
363                                     }))
364     ## create a white black other indicator
365     data$WhiteBlack ← FactorReduce(data$Race, tokeep = c("Black", "White", "U"))
366     data$DefWhiteBlack ← FactorReduce(data$DefRace, tokeep = c("Black", "White",
367                                                                "U"))
368     data$VicWhiteBlack ← FactorReduce(data$VictimRace, tokeep = c("Black", "White",
369                                                                "U"))
370     ## return the data with synthesized columns
371     data
372 }
373
374 ## write functions to process the sentences
375 SentenceProcess ← function(sentencing) {
376     sents ← tolower(sentencing)
377     ## identify sentences in months, years, and days
378     monthsent ← str_extract(sents, "[0-9\\-]+\\s*(?=m)")
379     daysent ← str_extract(sents, "[0-9\\-]+\\s*(?=d)")
380     yearsent ← str_extract(sents, "[0-9\\-]+\\s*(?=y)")
381     ## extract life without parole
382     lwp ← str_extract(sents, "parol[e]*")
383     ## and with parole
384     life ← str_extract(sents, "life")
385     life[!is.na(lwp)] ← NA
386     ## get restitutions
387     resti ← str_extract(sents, "[0-9,]+\\s*(?=restitu)|\\$[0-9,]+")
388     ## get supervised probation
389     supprob ← str_extract(sents, "sup.*pro")
390 }
391
392 ## Summary Functions #####
393 ## make a function to summarize trial jury data
394 JurySummarize ← function(Varnames = c("Disposition", "Race", "Gender", "
395     PoliticalAffiliation")) {
396     ## check if a juror summary object exists already
397     if (!("sun.juror" %in% ls(.GlobalEnv))) {
398         ## first group the data for easy access
399         Juries ← aggregate(sun.swap[, Varnames],
400                             by = list(TrialNumberID = sun.swap$TrialNumberID,
401                                     JurorNumer = sun.swap$JurorNumber),
402                             unique)

```

```

395 } else Juries ← sun.juror
396 ## in either case, perform aggregation by trial instance
397 Juries ← aggregate(Juries[, Varnames],
398                   by = list(TrialNumberID = Juries$TrialNumberID),
399                   function(var) var)
400 ## clean up the names
401 names(Juries)[grepl("Polit", names(Juries))] ← "PolAff"
402 Varnames[4] ← "PolAff"
403 ## now summarize relevant features
404 Summary ← apply(Juries[, Varnames], 1,
405                function(row) {
406                  ## get final jury indices
407                  disps ← unlist(row$Disposition)
408                  foreman ← grepl("Foreman", disps)
409                  finJur ← grepl("Foreman|Kept", disps)
410                  defStruck ← grepl("D_rem", disps)
411                  proStruck ← grepl("S_rem", disps)
412                  ## process all variables
413                  newrow ← sapply(row,
414                                function(el) {
415                                  c(Jury = table(unlist(el)[finJur]),
416                                    Venire = table(unlist(el)),
417                                    DefRem = table(unlist(el)[
418                                      defStruck]),
419                                    ProRem = table(unlist(el)[
420                                      proStruck]))
421                                })
422                  newrow$Disposition ← NULL
423                  newrow ← c(unlist(newrow), ForeRace = row$Race[foreman],
424                            ForeGender = row$Gender[foreman], ForePol =
425                              row$PolAff[foreman])
426                  if (sum(foreman) > 1) {
427                    names(newrow)[names(newrow) == "ForeRace1"] ← "
428                      ForeRace"
429                    names(newrow)[names(newrow) == "ForeGender1"] ← "
430                      ForeGender"
431                    names(newrow)[names(newrow) == "ForePol1"] ← "
432                      ForePol"
433                  }
434                  newrow
435                })
436 ## perform some clean up
437 longest ← sapply(Summary, length)
438 longest ← which(longest == max(longest))[1]
439 longNames ← names(Summary[[longest]])
440 Summary ← lapply(names(Summary[[longest]]),
441                 function(name) unname(sapply(Summary,
442                                               function(el) el[name])))
443 names(Summary) ← longNames
444 Summary ← lapply(longNames,
445                 function(nm) {
446                   if (grepl("ForeGender", nm)) {
447                     Summary[[nm]] ← factor(Summary[[nm]], levels = 1:3,
448                                           labels = LevGen)
449                   } else if (grepl("ForePol", nm)) {
450                     Summary[[nm]] ← factor(Summary[[nm]], levels = 1:5,
451                                           labels = LevPol)
452                   } else if (grepl("ForeRace", nm)) {
453                     Summary[[nm]] ← factor(Summary[[nm]], levels = 1:7,
454                                           labels = LevRace)
455                   } else Summary[[nm]]
456                 })
457 names(Summary) ← longNames
458 ## return these
459 list(Juries = Juries, Summaries = as.data.frame(Summary))
460 }
461 ## a generic simplification method to summarize a vector
462 Simplifier ← function(col, ...) {
463   UseMethod("Simplifier")
464 }

```

```

456 }
457
458 ## code up methods for the types to be seen
459 Simplifier.default ← function(col, collapse = "") paste0(col, collapse = collapse
460 )
461 Simplifier.numeric ← function(col, na.rm = TRUE, trim = 0, ...) mean.default(col,
462   trim = trim, na.rm = na.rm)
463 Simplifier.factor ← function(col, collapse = "", ...) paste0(sort(as.character(
464   levels(col)[as.numeric(col)])),
465   collapse = collapse
466 )
467 Simplifier.character ← function(col, collapse = "", ...) paste0(sort(col),
468   collapse = collapse)
469
470 ## create a grouping wrapper which does unique aggregation of a data set
471 UniqueAgg ← function(data, by, ...) {
472   ## convert data to a data frame for regularity
473   if (!is.data.frame(data)) data ← as.data.frame(data)
474   ## identify the grouping column by in the data
475   by.groups ← names(data) == by
476   ## provide nice error handling
477   stopifnot(sum(by.groups) > 0)
478   ## first identify which rows are already unique
479   groups ← as.numeric(as.factor(unlist(data[by.groups])))
480   unqRows ← sapply(groups, function(el) sum(groups == el) == 1)
481   ## consider grouping only the other rows using the unique function
482   endata ← data[unqRows,]
483   unqdata ← aggregate(data[!unqRows, !by.groups], by = list(data[!unqRows, by.
484     groups]), unique)
485   ## reorder to make sure everything is compatible
486   names(unqdata)[1] ← by
487   unqdata ← unqdata[,match(names(endata), names(unqdata))]
488   ## now use the Simplifier helper defined above to process these results
489   procddata ← lapply(unqdata, function(col) simplify(col, Simplifier, ...))
490   ## append everything together
491   endata ← lapply(1:length(endata),
492     function(n) c(if (is.factor(endata[[n]])) as.character(
493       endata[[n]] else endata[[n]],
494       procddata[[n]]))
495   names(endata) ← names(data)
496   ## convert to a data frame
497   as.data.frame(endata)
498 }
499
500 ## a simple helper to convert multiple factor levels into a single 'other' level
501 FactorReduce ← function(vals, tokeep) {
502   chars ← as.character(vals)
503   ## simply replace elements
504   chars[!grepl(paste0(tokeep, collapse = "|"), chars)] ← "Other"
505   chars
506 }
507
508 ## write a function to re-level factor variables to make mosaic plots cleaner
509 MatRelevel ← function(data) {
510   temp ← lapply(data, function(el) if (is.factor(el)) as.factor(levels(el)[as.
511     numeric(el)] else el)
512   temp ← as.data.frame(temp)
513   names(temp) ← names(data)
514   temp
515 }
516
517 ## another simple processing function to correct NA's given some other identifier
518 and data set
519 FillNAs ← function(dataNAs, filldata, identifier) {
520   ## extract the relevant column indices in a flexible way
521   if (is.null(colnames(filldata))) {
522     relcol ← grepl(identifier, names(filldata))
523   } else relcol ← grepl(identifier, colnames(filldata))
524   ## first identify the relevant rows in the data NAs
525   relRows ← is.na(dataNAs)

```

```

517     ## take the relevant rows of the filldata
518     filldata <- matrix(unlist(filldata[relcol]), ncol = sum(relcol))
519     rowfiller <- rowSums(filldata[relRows,])
520     ## return the filled data
521     dataNAs[relRows] <- rowfiller
522     dataNAs
523 }
524
525 ## write a wrapper to estimate the values of total removed jurors
526 RemovedJurorEstimates <- function(tofill, data, ident, plot = TRUE) {
527   temp <- FillNAs(tofill, filldata = data, identifier = ident)
528   temp2 <- rowSums(data[,grepl(ident, names(data))])
529   ## let's see how accurate this is if plotting is desired
530   if (plot) {
531     plot(temp, temp2, xlab = "Observed and Filled", ylab = "Juror Sums")
532     abline(0,1)
533   }
534   cat("= : ", sum(temp == temp2)/length(temp2), "\n", "< : ", sum(temp2 < temp)
535     /length(temp2), "\n", sep = "")
536   ## replace the filled values less than the estimated, for consistency
537   temp[temp < temp2] <- temp2[temp < temp2]
538   temp
539 }
540
541 ## LOADING AND PROCESSING DATA #####
542
543 ## load the data
544 SunshineData <- lapply(SunshineSheets, function(nm) as.data.frame(read_excel(
545   SunshineFile, sheet = nm)))
546 names(SunshineData) <- SunshineSheets
547 NorthCarData <- read.csv(NorthCarFile)
548 PhillyData <- read.csv(PhillyFile)
549
550 ## clean non-informative columns
551 CleanSunshine <- lapply(SunshineData, function(dat) dat[, !apply(dat,2,function(
552   col) all(is.na(col)))])
553
554 ## the Sunshine data needs to be restructured into one table, rather than a
555 ## relational database structure
556 ## see the IDMatch function, this was created specifically to perform ID-based
557 ## table joins
558 ## the most appropriate global target is the juror table, start by matching this
559 ## to the trial
560 FullSunshine <- with(CleanSunshine, CleaningMerge(Jurors, Trials, by = "
561   TrialNumberID"))
562 ## remove extra ID column, fix a misleading name
563 FullSunshine$CountyName <- FullSunshine$CountyID
564 FullSunshine$CountyID <- NULL
565 ## clean up two additional columns which had inconsistencies
566 FullSunshine$Disposition <- toupper(FullSunshine$Disposition)
567 FullSunshine$Race[FullSunshine$Race == "?"] <- "U"
568 ## before appending everything to this table, perform some other joins
569 TrialsToCharge <- with(CleanSunshine, CleaningMerge(Charges, Junction, by = "
570   ACISID", all = TRUE))
571 DefendantToTrial <- with(CleanSunshine, CleaningMerge(Defendants, DefendantTrial,
572   by = "DefendantID", all = TRUE))
573 AttorneyToTrial <- with(CleanSunshine, CleaningMerge(Attorney, AttorneyTrial, by =
574   "DefAttyID", all = TRUE))
575 ProsecutorToTrial <- with(CleanSunshine, CleaningMerge(Prosecutor, ProsecutorTrial
576   , by = "ProsecutorID", all = TRUE))
577 ## merge issues:
578 ##   - trials to charge: one charge is missing a trial ID, hopefully not
579 ##     important
580 ##   - prosecutors to trials: 26 prosecutors without trials, however all entries
581 ##     were entirely uninformative
582 ## given the above outputs, rename the failed clean merges to make the next
583 ## section cleaner
584 TrialsToCharge <- TrialsToCharge$Merge
585 ProsecutorToTrial <- ProsecutorToTrial$Merge

```

```

573
574 ## now perform some additional merges to create one sheet/data.frame
575 ## add the judge descriptions (no issues)
576 FullSunshine ← CleaningMerge(FullSunshine, CleanSunshine$Judges, by = "JudgeID",
    all = TRUE)
577 ## the charges
578 FullSunshine ← CleaningMerge(FullSunshine, TrialsToCharge, by = "TrialNumberID",
    all = TRUE)
579 ## this leads to 22 jurors in trials without charges and 29 charges without
trials, inspecting these:
580 ## - the jurors without charges are all related to a trial with ID number
"710-01", thankfully the other data
581 ## for this case is complete, and so it may still be useful for viewing
jury behaviour
582 ## - the charges without trials are all of the form "710-0xx", suggesting the
omission of entire trials of some
583 ## relation, hopefully these were not too similar, or this exclusion can be
explained later
584 FullSunshine ← FullSunshine$Merge
585 ## the defendants
586 FullSunshine ← CleaningMerge(FullSunshine, DefendantToTrial, by = "TrialNumberID"
    , all = TRUE)
587 ## the attorneys
588 FullSunshine ← CleaningMerge(FullSunshine, AttorneyToTrial, by = "TrialNumberID",
    all = TRUE)
589 ## the prosecutors
590 FullSunshine ← CleaningMerge(FullSunshine, ProsecutorToTrial, by = "TrialNumberID"
    , all = TRUE)
591 ## 26 jurors appear to be lacking a prosecutor, these appear to be the
uninformative prosecutors from earlier, included
592 ## due to the preferential inclusion of the missing values in the first of the
merged matrices
593 FullSunshine ← FullSunshine$Merge
594
595 ## perform some cleanup
596 ## start with some specific factor replacements
597 ## replace the "N" with "I", as these factor levels are interchangeable in the
codebook and prevent confusion with race
598 FullSunshine[,grepl("Pol", names(FullSunshine))] ← lapply(FullSunshine[,grepl("
    Pol", names(FullSunshine))],
599
600                                     function(var) {
601                                     var ← toupper(var)
602                                     var[var == "N"] ←
603                                     "I"
604                                     var
605                                     })
606 ## next save most variables as factors
607 FullSunshine ← lapply(FullSunshine,
608                       function(el) if (is.character(el)) as.factor(el) else el)
609 ## correct some overzealous assignment from above
610 FullSunshine[,grepl("Notes", names(FullSunshine))] ← lapply(FullSunshine[,grepl("
    Notes", names(FullSunshine))],
611
612                                     as.character)
613 ## perform factor regularization according to the factor levels provided in the
codebook
614 FullSunshine ← sapply(FullSunshine,
615                       function(el) {
616                       if (!is.factor(el)) {
617                       el[el == 999] ← NA
618                       el
619                       } else {
620                       el ← as.character(el)
621                       el ← toupper(el)
622                       el[is.na(el)] ← "U"
623                       as.factor(el)
624                       }
625                       }, simplify = FALSE)
626 FullSunshine ← as.data.frame(FullSunshine)
627 ## remove some unnecessary columns
628 FullSunshine$ID ← NULL

```

```

626 FullSunshine$TrialIDAuto ← NULL
627 ## combine the name columns to produce more useful columns
628 FullSunshine$JName ← paste(FullSunshine$JFirstName, FullSunshine$JLastName)
629 FullSunshine$JName[FullSunshine$JName == "U U"] ← "U"
630 FullSunshine$DefAttyName ← paste(FullSunshine$DCFirstName, FullSunshine$
    DCLastName)
631 FullSunshine$DefAttyName[FullSunshine$DefAttyName == "U U"] ← "U"
632 FullSunshine$ProsName ← paste(FullSunshine$ProsecutorFirstName, FullSunshine$
    ProsecutorLastName)
633 FullSunshine$ProsName[FullSunshine$ProsName == "U U"] ← "U"
634
635 ## Checkpoint 1: the clean data has been processed, none of the swaps, synthesis,
    or expansion has taken place
636 ## save this
637 if (!("FullSunshine.csv" %in% list.files())) write.csv(FullSunshine, "
    FullSunshine.csv", row.names = FALSE)
638 ## load if the desire is to start at checkpoint 1
639 if (!("FullSunshine" %in% ls())) FullSunshine ← read.csv("FullSunshine.csv")
640
641 ## Note: the below swap functions have been set to auto as the function's
    performance in these cases has already
642 ## been assessed, and so the swaps have already been inspected, it is critical
    for new data that "auto" be switched
643 ## off to take full advantage of this functionality, and so the wrapper "
    SwapandError" should not be used
644 ## in the juror data
645 sun.swapJuror ← SwapandError(FullSunshine, CorrectLevs = list(Race = LevRace,
646     Gender = LevGen,
647     PoliticalAffiliation
        = LevPol))
648
649 ## in the judge data
649 sun.swap ← SimpleSwapper(sun.swapJuror, CorrectLevs = list(JRace = LevRace,
650     JGender =
        LevGen,
651     JPoliticalAff
        = LevPol
        ))
652
653 ## viewing the error report of these data, they are all related to one judge,
    Arnold O Jones II, who is verified
654 ## as a male after a quick Google search
654 unique(sun.swap$Data[sun.swap$Errors, c("JFirstName", "JLastName")])
655 sun.swapJudge ← sun.swap$Data
656 sun.swapJudge$JGender[sun.swap$Errors] ← "M"
657 sun.swapJudge$JGender ← as.factor(levels(sun.swapJudge$JGender)[as.numeric(sun.
    swapJudge$JGender)])
658 ## in the prosecutor data
659 sun.swap ← SimpleSwapper(sun.swapJudge, CorrectLevs = list(ProsRace = LevRace,
660     ProsGender =
        LevGen,
661     ProsPoliticalAff
        = LevPol
        ))
662
663 ## that found no errors
663 ## a quick check of the levels of the defendant data finds only one error
664 levels(sun.swap$DefGender)
665 levels(sun.swap$DefRace)
666 sun.swap ← SwapandError(sun.swap, CorrectLevs = list(DefRace = LevRace,
667     DefGender = LevGen
        ))
668
669 ## next the attorney data
669 sun.swap ← SwapandError(sun.swap, CorrectLevs = list(DCRace = LevRace,
670     DCGender = LevGen,
671     DCPoliticalAff
        = LevPol))
672
673 ## finally the victim data
673 sun.swap ← SwapandError(sun.swap, CorrectLevs = list(VictimRace = LevRace,
674     VictimGender =
        LevGen))
675
676 ## this leaves the data error-free (in at least the race/gender/politics columns)

```



```

677 ## fix the outcome data, which had some improper levels
678 sun.swap$Outcome[sun.swap$Outcome == "HC"] ← "U"
679 sun.swap$Outcome[sun.swap$Outcome == "G"] ← "GC"
680 sun.swap$Outcome ← as.factor(levels(sun.swap$Outcome)[as.numeric(sun.swap$Outcome
681   )])
682 ## lets make the levels more clear for some of the data (race, politics,
683   disposition)
684 ## start with the disposition
685 levels(sun.swap$Disposition) ← c("C_rem", "D_rem", "Foreman", "Kept", "U_rem",
686   "S_rem", "Unknown")
687 ## next the political affiliation
688 sun.swap ← lapply(sun.swap, function(el) {
689   if (is.factor(el) & identical(levels(el), LevPol)) {
690     levels(el) ← c("Dem", "Ind", "Lib", "Rep", "U")
691   } else el})
692 levels(sun.swap$JPoliticalAff) ← c("Dem", "Ind", "Rep", "U")
693 ## now the race
694 sun.swap ← lapply(sun.swap, function(el) {
695   if (is.factor(el) & identical(levels(el), LevRace)) {
696     levels(el) ← c("Asian", "Black", "Hisp", "NatAm", "Other",
697       "U", "White")
698   } else el})
699 levels(sun.swap$VictimRace) ← c("Asian", "Black", "Hisp", "NatAm",
700   "U", "White")
701 levels(sun.swap$JRace) ← c("Black", "Hisp", "NatAm", "U", "White")
702 levels(sun.swap$DCRace) ← c("Asian", "Black", "NatAm", "Other",
703   "U", "White")
704 ## now the outcome/verdict
705 levels(sun.swap$Outcome) ← c("Acquittal", "Guilty as Charged",
706   "Guilty of Lesser", "Incomplete", "Mistrial",
707   "U")
708 ## the defense attorney type
709 levels(sun.swap$DefAttyType) ← c("App Priv", "Public", "Private",
710   "Ret Priv", "U", "Waived")
711 ## add a guilt indicator
712 sun.swap$Guilty ← grepl("Guilty", sun.swap$Outcome)
713 ## add a simple indicator of defendant race matching juror race if they are both
714   known
715 sun.swap$RaceMatch ← sun.swap$Race == sun.swap$DefRace
716 sun.swap$RaceMatch[sun.swap$Race == "U" | sun.swap$DefRace == "U"] ← NA
717 ## now perform tree classification of crimes
718 ## first cast sun.swap as a data frame
719 sun.swap ← as.data.frame(sun.swap)
720 ## regularize the charges
721 chargFact ← as.factor(sun.swap$ChargeTxt)
722 regCharg ← StringReg(levels(chargFact)[as.numeric(chargFact)])
723 ## classify these into a charge tree and aggregate this at the coarsest level
724 aggCharg ← treeAgg(stringTree(regCharg, chargeTree))
725 ## these can be further classified into crime classes
726 crimes.trial ← CrimeClassify(aggCharg, regCharg)
727 ## convert these classes into a factor for the data, start with a generic "other"
728   vector
729 sun.swap$CrimeType ← rep("Other", nrow(sun.swap))
730 ## now populate it
731 for (nm in sort(names(crimes.trial))) sun.swap$CrimeType[crimes.trial[[nm]]] ← nm
732 sun.swap$CrimeType ← as.factor(sun.swap$CrimeType)
733 ## synthesize additional columns
734 sun.swap ← SynCols(sun.swap)
735 ## now organize this on the juror scale
736 sun.juror ← UniqueAgg(sun.swap, by = "JurorNumber", collapse = ",")
737 ## Checkpoint 2: the swapped data has been processed and summarized to be on the

```

```

    scale of individual jurors
743 ## save the swapped data
744 write.csv(sun.swap, "FullSunshine_Swapped.csv", row.names = FALSE)
745 ## and the juror summarized data
746 saveRDS(sun.juror, "JurorAggregated.Rds")
747
748 ## summarize by trial, get the unique trials
749 Trials <- unique(sun.swap$TrialNumberID)
750 ## extract information about these trials, note that grouping occurs on the trial
    ID, defendant ID, and charge ID levels,
751 ## as the trials frequency involve multiple charges and defendants, which makes
    them less clean
752 sun.trial <- aggregate(sun.swap[, TrialVars],
753                        by = list(sun.swap$TrialNumberID, sun.swap$DefendantID
                                .DefendantToTrial,
                                sun.swap$ID.Charges),
754                        unique)
755
756 sun.trial$Group.1 <- NULL
757 sun.trial$Group.2 <- NULL
758 sun.trial$Group.3 <- NULL
759
760 ## summarize the juries by trial as well
761 sun.jursum <- JurySummarize()
762
763 ## merge the summaries to the trial sunshine data
764 sun.trialsum <- merge(cbind(TrialNumberID = sun.jursum$Juries$TrialNumberID, sun.
    jursum$Summaries),
765                      sun.trial, all = TRUE)
766
767 ## notice that the total removed variables are incomplete, try to correct this
    where possible using the jury
768 ## summarized data above
769 sun.trialsum$DefRemEst <- RemovedJurorEstimates(sun.trialsum$DefenseTotalRemoved,
    data = sun.trialsum,
770                                                ident = "Gender.DefRem", plot =
    FALSE)
771 ## perform this same procedure for the prosecution removals
772 sun.trialsum$ProRemEst <- RemovedJurorEstimates(sun.trialsum$StateTotalRemoved,
    data = sun.trialsum,
773                                                ident = "Gender.ProRem", plot =
    FALSE)
774 ## synthesize some other variables, simple race indicators
775 sun.trialsum$DefWhiteBlack <- as.factor(FactorReduce(sun.trialsum$DefRace, tokeep
    = c("Black", "White", "U")))
776 sun.trialsum$DefWhiteOther <- as.factor(FactorReduce(sun.trialsum$DefWhiteBlack,
    tokeep = c("White", "U")))
777 ## the Kullback-Leibler divergence
778 sun.trialsum$KLdiv <- kldiv(sun.trialsum[, grepl("Jury", names(sun.trialsum))],
    sun.trialsum[, grepl("Venire", names(sun.trialsum))])
779
780
781 ## Checkpoint 3: the data has been set to the trial level and summarized
782 ## save this
783 saveRDS(sun.trialsum, "TrialAggregated.Rds")
784 saveRDS(sun.jursum, "AllJuries.Rds")
785
786
787 ## CHARGE CLASSIFICATION IMAGES #####
788 ## presented here is the code to generate the appendix charge classification
    images
789
790 ## extract relevant charges from the clean sunshine data, avoiding duplicates
791 ## regularize the charges
792 chargFact.clean <- as.factor(CleanSunshine$Charges$ChargeTxt)
793 regCharg.clean <- StringReg(levels(chargFact.clean))[as.numeric(chargFact.clean)]
794 ## classify these into a charge tree and aggregate this at the coarsest level
795 tree.clean <- stringTree(regCharg.clean, chargeTree)
796
797 ## define padding
798 crimepad <- 0.01
799 areatop <- 0.94

```

```

800 hght ← ((0.94 - 0.035) - 5*crimepad)/5
801 wid ← ((0.975 - 0.025) - 7*crimepad)/6
802
803 ## add overall box
804 grid.newpage()
805 grid.rect(width = 0.95, height = 0.95)
806 grid.text(label = "Charges", x = 0.025 + crimepad/2, y = 0.975 - crimepad/2, just
      = c("left","top"))
807 grid.text(label = "(198)", x = 0.975 - crimepad/2, y = 0.975 - crimepad/2, just =
      c("right","top"))
808
809 ## inner crime boxes
810 xpos ← crimepad + 0.025 + wid/2 + (0:5)*(crimepad + wid)
811 ypos ← crimepad + 0.025 + hght/2 + (0:4)*(crimepad + hght)
812 coords ← cbind(rep(xpos, each = 5), rev(ypos))[1:length(tree.clean)-1,]
813 grid.rect(x = coords[,1], y = coords[,2], width = wid, height = hght)
814 grid.text(label = names(tree.clean)[1:(length(tree.clean)-1)], x = coords[,1] - (
      wid - crimepad)/2,
      y = coords[,2] + (hght-crimepad)/2, just = c("left","top"))
815
816
817 ## add unclassified and top level counts
818 unclass ← sapply(tree.clean, function(el) if (is.list(el)) length(el$other) else
      0)
819 grid.text(label = paste0("(", unclass, ")"), x = coords[,1] + (wid - crimepad)/2,
      y = coords[,2] + (hght-crimepad)/2,
      just = c("right", "top"))
820
821 toplev ← sapply(tree.clean, function(el) if (is.list(el)) "" else length(el))
822 grid.text(label = toplev, x = coords[,1], y = coords[,2])
823
824 ## add sublist counts
825 for (ii in 1:length(tree.clean)) {
826   if (is.list(tree.clean[[ii]])) {
827     currlist ← tree.clean[[ii]]
828     len ← length(currlist) - 1
829     boty ← coords[ii,2] - hght/2
830     newwid ← wid - 2*crimepad
831     newhght ← (hght - 0.035 - len*crimepad)/len
832     newy ← crimepad + newhght/2 + (0:(len-1))*(newhght + crimepad) + boty
833     if (names(tree.clean)[ii] == "assa") {
834       grid.rect(x = coords[ii,1], y = newy, width = newwid, height =
         newhght)
835       grid.text(label = names(currlist), x = coords[ii,1] - (newwid-
         crimepad)/2, y = newy,
         just = "left", gp = gpar(fontsize = 8))
836       grid.text(sapply(currlist, length), x = coords[ii,1], y = newy, gp =
         gpar(fontsize = 8))
837     } else if (names(tree.clean)[ii] == "murder") {
838       grid.rect(x = coords[ii,1], y = newy, width = newwid, height =
         newhght)
839       grid.text(label = names(currlist), x = coords[ii,1] - (newwid-
         crimepad)/2, y = newy + (newhght-crimepad)/2,
         just = c("left","top"), gp = gpar(fontsize = 8))
840       grid.text(paste0("(", sapply(currlist[1:len], function(el) length(el$
         other)), ")"), x = coords[ii,1] + (newwid-crimepad)/2,
         y = newy + (newhght-crimepad)/2, just = c("right","top"),
         gp = gpar(fontsize = 8))
841       grid.rect(x = coords[ii,1], y = newy - 0.01, width = newwid - 2*
         crimepad, height = (newhght - 0.02)/2)
842       grid.text(rep("att",2), x = coords[ii,1] - (newwid-2*crimepad-
         crimepad)/2, y = newy-0.01,
         just = "left", gp = gpar(fontsize = 8))
843       grid.text(sapply(currlist[1:len], function(el) length(el$att)), x =
         coords[ii,1],
         y = newy - 0.01, gp = gpar(fontsize = 8))
844     } else {
845       grid.rect(x = coords[ii,1], y = newy, width = newwid, height =
         newhght)
846       grid.text(label = names(currlist), x = coords[ii,1] - (newwid-
         crimepad)/2, y = newy + (newhght-crimepad)/2,
         just = c("left","top"), gp = gpar(fontsize = 8))
847     }
848   }
849 }
850
851
852

```

```

853         grid.text(sapply(currlist, length), x = coords[ii,1], y = newy, gp =
            gpar(fontsize = 10))
854     }
855 }
856 }
857
858 ## a small example tree
859 firstx ← 0.33
860 secondx ← 0.75
861 firsty ← c(0.25,0.75)
862 secondy ← c(0.125,0.375,0.625,0.875)
863 wd ← 0.3
864 hg ← 0.1
865 grid.newpage()
866 grid.rect(x = firstx, y = firsty, width = wd, height = hg)
867 grid.rect(x = secondx, y = secondy, width = wd, height = hg)
868 grid.lines(x = c(0,firstx-wd/2), y = c(0.5,firsty[1]))
869 grid.lines(x = c(0,firstx-wd/2), y = c(0.5,firsty[2]))
870 grid.lines(x = c(firstx+wd/2,secondx-wd/2), y = c(firsty[2],secondy[3]))
871 grid.lines(x = c(firstx+wd/2,secondx-wd/2), y = c(firsty[2],secondy[4]))
872 grid.lines(x = c(firstx+wd/2,secondx-wd/2), y = c(firsty[1],secondy[1]))
873 grid.lines(x = c(firstx+wd/2,secondx-wd/2), y = c(firsty[1],secondy[2]))
874 grid.text(label = c("sex(?.*offend)", "sex(?.*offense)"), x = firstx, y = firsty
            ,
875         gp = gpar(fontsize = 16))
876 grid.text(label = c("first|1", "second|2", "regis", "addr"), x = secondx, y = rev(
            secondy),
877         gp = gpar(fontsize = 16))

```

D.2 Analysis Code

As for the data processing code, this code is more conveniently accessed using the author's GitHub (https://github.com/Salahub/peremptory_challenges).

```

1 #####
2
3 ## THESIS ANALYSIS SCRIPT
4 ## Christopher Salahub
5 ## Sept 26, 2018
6
7 #####
8
9 ## PACKAGES #####
10 library(readxl)
11 library(MASS)
12 library(eikosograms)
13 library(RColorBrewer)
14 library(stringr)
15 library(tm)
16 library(lme4)
17 library(lmerTest)
18
19 ## CONSTANTS #####
20
21 ## start by defining file locations
22 ThesisDir ← "c:/Users/Chris/Documents/ETH Zurich/Thesis/Data"
23 SunshineFile ← paste0(ThesisDir, "/JurySunshineExcel.xlsx")
24 SunshineSheets ← excel_sheets(SunshineFile)
25
26 NorthCarFile ← paste0(ThesisDir,
27     "/Jury Study Data and Materials/NC Jury Selection Study
28     Database6 Dec 2011.csv")
29
30 PhillyFile ← paste0(ThesisDir,
31     "/Voir Dire Data & Codebook/capital_venires.csv")

```

```

31
32 ## next the factor level codes as given in the codebook and regularized here
33 ## regularization: - political affiliation "N" replaced with "I" for all entries
34 LevRace ← sort(c("Asian", "Black", "Hispanic", "NatAm", "Other", "U", "White"))
35 LevGen ← sort(c("F", "M", "U"))
36 LevPol ← sort(c("Dem", "Lib", "Rep", "Ind", "U"))
37
38 ## color constants
39 racePal ← brewer.pal(3, "Set2") # c("steelblue", "grey50", "firebrick")
40 whitePal ← c("steelblue", "firebrick")
41 crimePal ← brewer.pal(7, "Set1")
42 dispPal ← brewer.pal(3, "Set2")
43
44
45 ## FUNCTIONS #####
46
47 ## create a plot which visualizes positional data patterns by a categorical
48 ## variable
49 ## could encode density as either box sizes or through alpha levels of colour
50 ## the areal representation breaks the "dimensionality" rule of data in Edward
51 ## Tufte's "The Visual Display of Information",
52 ## to limit the dimensionality of a representation to at most the dimensionality
53 ## of the data itself
54 ## place the legend labels in the largest box instead of off to the side (didn't
55 ## really work...)
56 posboxplot ← function(x, y, cats, boxcolours = NULL, boxwids = 0.8, alphaencoding
57 = TRUE, alphamin = 0.1,
58 areaencoding = FALSE, xlim = range(x) + boxwids*c(-1/
59 1.05, 1/1.05), inc.leg = TRUE,
60 ylim = range(y) + boxwids*c(-1/1.05, 1/1.05), ...) {
61   ## extract the number of categories to be displayed in each small multiple
62   ncats ← length(unique(cats))
63   ## automatically generate the category colours using color brewer
64   if (is.null(boxcolours)) {
65     boxcols ← brewer.pal(ncats, "Set2")
66     boxcolours ← boxcols
67   } else boxcols ← boxcolours
68   ## first identify the unique coordinates for the small multiples
69   unqPos ← unique(cbind(x, y))
70   ## iterate through these, create tables of the categories at each position
71   cattabs ← t(apply(unqPos, 1, function(pos) {
72     ## generate a count a table
73     table(cats[x == pos[1] & y == pos[2]])
74   }))
75   ## sum these counts to get the total at each position to scale the small
76   ## multiples
77   rowcounts ← rowSums(cattabs)
78   ## convert the count table to cumulative proportions at each position
79   catprops ← t(apply(cbind(0, cattabs/rowcounts), 1, cumsum))
80   ## use these and the table of counts to generate the small multiples
81   ## first in the case that opacity encodes density
82   if (alphaencoding) {
83     ## in the opacity-density case, convert the box colours to rgb and
84     ## replicate them as necessary
85     boxcols ← col2rgb(rep(boxcols, each = nrow(catprops)), alpha = FALSE)/255
86     ## convert back to hex, adding the alpha encoding to control opacity
87     boxcols ← rgb(t(boxcols),
88       alpha = rep(round((1-alphamin)*rowcounts/max(rowcounts) +
89         alphamin, digits = 4), times = ncats))
90     ## in the case size encodes density, simply replicate the colors for the
91     ## number of positions
92   } else boxcols ← rep(boxcols, each = nrow(catprops))
93   ## create an empty plot to place the small multiples
94   plot(x, y, col = NA, xlim = xlim, ylim = ylim, ...)
95   ## determine the width of the small multiple boxes
96   if (areaencoding) boxwids ← boxwids*sqrt(rowcounts/max(rowcounts))
97   ## define the bottom corner positions of the boxes
98   bottomx ← unqPos[,1] - boxwids/2
99   bottomy ← unqPos[,2] - boxwids/2
100   ## use the bottom corner positions to calculate box extents, with internal

```

```

    borders defined as well
91 rectx ← bottomx + catprops*boxwids
92 recty ← cbind(rep(bottomy, times = ncats), rep(bottomy + boxwids, times =
    ncats))
93 ## convert the x coordinates into a list of vectors specifying all positions
94 xvec ← lapply(1:(ncats+1), function(n) rectx[,n])
95 ## place the rectangles by unlisting this structure correctly
96 rect(xleft = unlist(xvec[1:ncats]), ybottom = recty[,1], xright = unlist(xvec
    [2:(ncats+1)]),
97     ytop = recty[,2], col = boxcols, border = boxcols)
98 ## include a legend if desired
99 if (inc.leg) legend(x = "top", legend = colnames(rectx)[-1], fill = boxcolours
    , bty = "n",
100                  xpd = NA, horiz = TRUE)
101 }
102
103 ## create a function for proportional parallel coordinate plots
104 ## incorporate possibility to display in a non-proportional absolute way
105 propparcoord ← function(fact, cats, levs = NULL, proportional = TRUE, includerel
    = proportional, ylim = NULL,
106                        colpal = NULL, ordering = NULL, legpos = "topleft",
                        brptpos = 1, brwid = 4, ...) {
107
108   ## create the x label
109   xnm ← deparse(substitute(fact))
110   ## perform a type check
111   if (!is.factor(fact)) fact ← as.factor(fact)
112   ## check if levs have been supplied
113   if (is.null(levs)) levs ← unique(cats)
114   ## extract the levels and indices of interest
115   levinds ← cats %in% levs
116   ## get the length of the categories provided and a table of frequencies
117   ctab ← table(as.character(cats[levinds]))
118   len ← length(fact)
119   lineLen ← length(levels(fact))
120   ## check if order is null and allow reordering
121   if (is.null(ordering)) ordering ← 1:lineLen
122   ## set the ylim and other values based on whether a proportional plot is
    desired
123   factab ← table(fact)
124   if (proportional) factab ← factab/len else if (is.null(ylim)) ylim ← c(0, max
    (factab))
125   ## generate a palette if one is not given
126   if (is.null(colpal)) colpal ← brewer.pal(length(ctab), "Set2")
127   colpal ← colpal[ordering]
128   ## check for missing ylim value
129   if (is.null(ylim)) yrng ← c(0,1) else yrng ← ylim
130   ## now plot everything
131   if (proportional) ynm ← "Proportion" else ynm ← "Count"
132   plot(NA, xlim = c(1,lineLen), ylim = yrng, xlab = xnm, ylab = ynm, xaxt = 'n'
    , ...)
133   lines(1:lineLen, factab[ordering])
134   ## create positions for relative proportions bar chart if this is desired
135   if (includerel) {
136     ## specify bar chart rectangle bounds
137     rectbounds ← seq(0.7, 0.7 - 0.03*(length(ctab)+3), by = -0.03)*yrng[2]
138     ## add the reference "total population" bar
139     rect(xleft = brptpos, xright = 1+brwid/4, ybottom = rectbounds[1], ytop =
    rectbounds[2], col = "black")
140   }
141   ## plot lines and relative size rectangles, depending on options
142   for (ii in 1:length(ctab)) {
143     ## get the counts for the subset selected by ii
144     subsetab ← table(fact[cats == names(ctab)[ii]])
145     ## set these proportional if desired
146     if (proportional) subsetab ← subsetab/ctab[names(ctab)[ii]]
147     ## place these in a line
148     lines(1:lineLen, subsetab[ordering], col = colpal[ii])
149     ## add the appropriate bar if desired
150     if (includerel) {
151       rect(xleft = brptpos, xright = 1+(brwid/4)*ctab[ii]/len, ybottom =

```

```

151         rectbounds[ii+1], ytop = rectbounds[ii+2],
152         col = colpal[ii])
153     }
154 }
155 ## add a legend and axis
156 legend(x = legpos, legend = c("All", names(ctab)), lty = 1, col = c("Black",
157     colpal), title = deparse(substitute(cats)))
158 if (includerel) text("Relative Totals", x = 1, y = 0.72*yrng[2], pos = 4)
159 axis(1, 1:lineLen, levels(fact)[ordering])
160 }
161 ## DEPRECATED: a specific case of the above function which has been replaced by
162 the following function
163 ## make a specific line plot function
164 parcoordrace <- function() {
165     ## clean up the defwhiteblack variable
166     DefWhiteBlack_clean <- as.factor(as.character(gsub(",Other|,U", "", sun.juror$
167         DefWhiteBlack)))
168     ## first generate the necessary mixed factor
169     jurorDef <- with(sun.juror, as.factor(paste(DefWhiteBlack_clean, WhiteBlack,
170         sep = ":")))
171     ## generate positions to associate these factor levels
172     xpos <- rep(1:5, each = 4) + rep(c(-0.21,-0.07,0.07,0.21), times = 5)
173     ## choose disposition levels
174     displevs <- c("", "Kept", "S_rem", "D_rem", "C_rem")
175     nicelevs <- c("All", "Jury", "Pros.", "Def.", "Cause")
176     ## create a table based on the mixed factor
177     mixtab <- with(sun.juror, lapply(displevs,
178         function(displ) {
179             tab <- table(jurorDef[grepl(displ,
180                 Disposition)])/sum(grepl(displ,
181                 Disposition))
182             wraptab <- c(tab,tab[1:4])
183             wraptab
184         })))
185     ## define a palette
186     colpal <- brewer.pal(length(displevs) - 1, "Set2")
187     ## extract the max value for plotting purposes
188     maxtab <- max(unlist(mixtab))
189     ## plot all tables using different colours
190     lapply(1:length(mixtab), function(ind) {
191         if (ind == 1) {
192             plot(x = xpos, y = mixtab[[ind]], xlim = range(xpos), ylim = c(0,
193                 maxtab), xlab = "", xaxt = "n", ylab = "Proportion of Data",
194                 col = "black", type = 'l', yaxt = 'n')
195         } else lines(xpos+0.006*(ind-4)+0.003, mixtab[[ind]], col = colpal[ind
196             -1], lty = 2)})
197     ## add axes
198     axis(side = 2, at = round(seq(0, maxtab, length.out = 3), digits = 2))
199     axis(1, at = xpos, labels = rep(c("Black","Other","Unknown","White"), times =
200         5))
201     axis(1, at = 1:5, labels = c("Black Defendant","Other","Unknown Defendant","
202         White Defendant","Black Defendant"),
203         pos = -0.05, xpd = NA, tick = FALSE)
204     ## add guide lines coloured by disposition
205     lapply(2:length(displevs),
206         function(ind) {
207             sapply(1:20, function(n) rect(xleft = rep(xpos[n],2)+0.006*(ind-4)
208                 , xright = rep(xpos[n],2)+0.006*(ind-3),
209                 ybottom = mixtab[[1]][n], ytop =
210                 mixtab[[ind]][n], border =
211                 colpal[ind-1],
212                 col = colpal[ind-1]))
213         })
214     ## add legend-ish text
215     text(x = xpos[1]-0.01, y = mixtab[[2]][1] + 0.0075, labels = nicelevs[2], pos
216         = 2, cex = 0.75, srt = 90,
217         col = colpal[1])
218     text(x = xpos[1]+0.01, y = mixtab[[3]][1]-0.02, labels = nicelevs[3], pos =
219         1, cex = 0.75, srt = 90, col = colpal[2])

```

```

205   text(x = xpos[1]+0.02, y = mixtab[[4]][1]+0.04, labels = nicelevs[4], pos =
      1, cex = 0.75, srt = 90, col = colpal[3])
206   text(x = xpos[1]+0.03, y = mixtab[[5]][1], labels = nicelevs[5], pos = 1, cex
      = 0.75, srt = 90, col = colpal[4])
207 }
208
209 ## the better version of the above function, takes an arbitrary three-way
      contingency table and plots the different conditional
210 ## probabilities of the desired margins
211 parcoordracev2 <- function(tabl = NULL, tracemar = 1, deslev = NULL, wid = 0.02,
      addlines = FALSE,
212                               space = 0.025, testlines = FALSE, ...) {
213   ## in the default case (no table provided), look at the key race
      relationships, as these motivated this study
214   if (is.null(tabl)) {
215     ## for cleanliness, remove those jurors with unknown races
216     temp.juror <- sun.juror[sun.juror$WhiteBlack != "U" & sun.juror$
      DefWhiteBlack != "U",]
217     temp.juror$WhiteBlack <- as.factor(as.character(temp.juror$WhiteBlack))
218     temp.juror$DefWhiteBlack <- as.factor(as.character(temp.juror$
      DefWhiteBlack))
219     ## perform the same operation for defendant race
220     temp.juror$DefWhiteBlack <- as.factor(as.character(gsub(",Other|,U", "",
      temp.juror$DefWhiteBlack)))
221     ## make a table of disposition and both races
222     outcometab <- table(temp.juror[,c("Disposition", "DefWhiteBlack", "
      WhiteBlack")])
223   } else { ## if a table is provided simply copy it to the internal "outcometab
      " argument
224     outcometab <- tabl
225   }
226   ## determine the "non-trace" margins; these specify margins which are
      combined to define the cases to which the horizontal
227   ## line segments correspond
228   nontrace <- (1:3)[1:3 != tracemar]
229   ## get the dimension names
230   tabnames <- dimnames(outcometab)
231   ## handle a null desired level setting
232   if (is.null(deslev)) deslev <- 1:length(tabnames[[tracemar]])
233   ## create a palette
234   temPal <- brewer.pal(length(deslev), "Set2")
235   ## calculate the conditional probability distribution of outcome given non-
      trace margins using outcometab
236   condout <- apply(outcometab, nontrace, function(margin) margin/sum(margin))
237   ## and the sums
238   marsums <- apply(outcometab, nontrace, sum)
239   ## extract the desired levels from the margin of interest, write this
      flexibly to allow programmatic margin extraction later
240   ## first define the local environment as the calling environment
241   evEnv <- environment()
242   ## create the language object
243   condoutinds <- condoutcall <- quote(condout[, ,])
244   ## place the levels of interest as the subset argument
245   condoutcall[[tracemar+2]] <- deslev
246   ## evaluate this to subset the data
247   condout <- eval(condoutcall, envir = evEnv)
248   ## rename some useful values for ease of reference
249   ## the dimensionality of the data
250   dims <- dim(condout)
251   ## the number of horizontal segments
252   nseg <- prod(dims[nontrace])
253   ## the number of 'inner' combination values
254   innern <- dims[nontrace[1]]
255   ## the number of 'outer' combination values (inner and outer refer to axis
      label positions)
256   outern <- dims[nontrace[2]]
257   ## add padding between segments to space everything nicely
258   ## first replicate the trace margin sums to create a vector of the correct
      size for the horizontal line generation
259   tempx <- rep(c(marsums), times = c(rep(2, nseg-1), 1))

```



```

260  ## replace every even element with the desired space size, then the
      cumulative sum automatically spaces
261  tempx[2*(1:(nseg-1))] ← space*rep(c(rep(1,innern-1),3), length.out = nseg-1)*
      sum(marsums)
262  ## take the cumulative sum to get the positions
263  xpos_line ← c(0, cumsum(tempx)/sum(tempx))
264  ## xpos ← rep(1:dims[nontrace[2]], each = dims[nontrace[1]]) +
265  ##   rep(seq(-0.2, 0.2, length.out = dims[nontrace[1]]), times = dims[
      nontrace[2]])
266  ## xpos ← cumsum(marsums)/sum(marsums)
267  ## create the empty plot region
268  plot(NA, xlim = range(xpos_line), ylim = c(0, max(condout[,])), xaxt = 'n',
      xlab = "",
269      ylab = "Conditional Probability", ...)
270  ## calculate and plot the horizontal lines at the mean values
271  ## the old way: calculating the mean of the conditional distributions
272  meanline ← apply(condout, nontrace, mean)
273  ## this way does not correspond to the hypothesis being tested: that there is
      no preference for race displayed by
274  ## either side, rather under this hypothesis, the expected rate of each
      combination is given by the product of the
275  ## overall rate for each side and the proportion of the venire of
276  for(ii in 1:length(meanline)) lines(c(xpos_line[2*ii-1],xpos_line[2*ii]), rep
      (meanline[ii],2))
277  ## add the vertical lines for each cell, save the positions for later
278  xpos ← sort(unlist(lapply(1:length(deslev), function(ind) {
279      ## first extract the relevant margin to give the y values
280      tempind ← condoutinds
281      tempind[[tracemar + 2]] ← ind
282      yvals ← eval(tempind, envir = evEnv)
283      ## use the horizontal line positions and the index to place the vertical
      lines
284      ## adjx ← xpos + wid*(ind - (1/2)*(1 + length(deslev)))
285      adjx ← xpos_line[2*(1:nseg)-1] + (ind-1)*diff(xpos_line)[2*(1:nseg)-1]/(
      dims[tracemar] - 1)
286      ## add the corresponding end points
287      points(adjx, yvals, col = temPal[ind], pch = 19)
288      ## for aesthetics exclude lines if confidence intervals are plotted
289      if (!testlines) for (ii in 1:length(adjx)) lines(x = rep(adjx[ii],2), y =
      c(meanline[ii], yvals[ii]),
290
      lty = 2, col = temPal[
      ind])
291      ## if trace lines (parallel axis plot) are desired plot these
292      if (addlines) lines(adjx, yvals, col = temPal[ind], lty = 3)
293      ## rect(xleft = adjx - (1/2)*wid, xright = adjx + (1/2)*wid, ybottom =
      meanline,
294      ##   ytop = yvals, col = temPal[ind])
295      return(adjx)
296  })))
297  ## now add the axes
298  ## first determine axis label positions
299  xpos ← sapply(1:nseg, function(ind) mean(xpos_line[c((2*ind-1),2*ind)]))
300  outrpos ← sapply(1:outern, function(ind) mean(range(xpos_line[(2*(ind-1)*
      innern + 1):(2*ind*innern)])))
301  ## add the axis ticks for the inner labels
302  axis(1, at = xpos, labels = rep("", length(xpos)))
303  ## add the labels to the inner axis
304  axis(1, at = xpos, tick = FALSE, labels = rep(tabnames[[nontrace[1]]], times
      = outern), cex.axis = 0.7,
305      pos = -0.02*max(condout))
306  ## add the labels for the outer axis
307  axis(1, at = outrpos, labels = tabnames[[nontrace[2]]], xpd = NA,
308      tick = FALSE, pos = -0.08*max(condout[,]))
309  ## provide the axis title to give context
310  axis(1, at = mean(range(xpos)), xpd = NA, tick = FALSE, pos = -0.15*max(
      condout),
311      labels = paste0("Inner label: ", names(tabnames)[nontrace[1]], " | Outer
      label: ", names(tabnames)[nontrace[2]]))
312  ## add testing lines if desired
313  if (testlines) {

```

```

314     ## get x positions
315     errpos ← xpos
316     ##errpos[3*(1:nseg) - 1] ← axpos
317     ## reformat y positions
318     erry ← c(condout)
319     ## define error bar extensions
320     ext ← c(-0.005, 0.005)
321     ## add bars at each position
322     invisible(sapply(1:length(erry), function(pos) {
323         ## rename the conditional probability for readability
324         p ← erry[pos]
325         ## extract the relevant margin count
326         n ← marsums[floor((pos-1)/dims[tracemar]) + 1]
327         ## calculate the binomial error size
328         err ← 2*sqrt((p/n)*(1-p))
329         ## and the appropriate colour
330         errcol ← temPal[(pos-1) %% dims[tracemar] + 1]
331         ## add vertical lines and horizontal end lines
332         lines(x = errpos[pos] + ext, y = rep(p + err, 2), col = adjustcolor(
            errcol, alpha.f = 0.5))
333         lines(x = errpos[pos] + ext, y = rep(p - err, 2), col = adjustcolor(
            errcol, alpha.f = 0.5))
334         lines(x = rep(errpos[pos], 2), y = c(p + err, p - err), col =
            adjustcolor(errcol, alpha.f = 0.5))
335         ##lines(, meanline - sqrt((meanline/(3*marsums))*(1-3*meanline)), lty
            = 2)
336         ##lines(xpos, meanline + sqrt((meanline/(3*marsums))*(1-3*meanline)),
            lty = 2)
337     })))
338 }
339 ## add a legen to explain the colours
340 legend(x = "top", horiz = TRUE, legend = tabnames[[tracemar]][deslev], col =
    temPal, inset = -0.04, cex = 0.7,
341       fill = temPal, bg = "white", xpd = NA)
342 ##invisible(sapply((0:4)*0.05, function(val) lines(x = c(0,max(xpos)+1), y =
    rep(val,2), col = "white", lwd = 2)))
343 }
344
345 ## the better version of the above function, takes an arbitrary three-way
    contingency table and plots the different conditional
346 ## probabilities of the desired margins
347 testplot ← function(tabl = NULL, tracemar = 1, deslev = NULL, wid = 0.02,
    addlines = FALSE,
348                   space = 0.025, testlines = FALSE, expected = NULL,
    ...) {
349     ## in the default case (no table provided), look at the key race
    relationships, as these motivated this study
350     if (is.null(tabl)) {
351         ## for cleanliness, remove those jurors with unknown races
352         temp.juror ← sun.juror[sun.juror$WhiteBlack != "U" & sun.juror$
            DefWhiteBlack != "U",]
353         temp.juror$WhiteBlack ← as.factor(as.character(temp.juror$WhiteBlack))
354         temp.juror$DefWhiteBlack ← as.factor(as.character(temp.juror$
            DefWhiteBlack))
355         ## perform the same operation for defendant race
356         temp.juror$DefWhiteBlack ← as.factor(as.character(gsub(",Other|,U", "",
            temp.juror$DefWhiteBlack)))
357         ## make a table of disposition and both races
358         outcometab ← table(temp.juror[,c("Disposition", "DefWhiteBlack", "
            WhiteBlack")])
359     } else { ## if a table is provided simply copy it to the internal "outcometab
        " argument
360         outcometab ← tabl
361     }
362     ## determine the "non-trace" margins; these specify margins which are
        combined to define the cases to which the horizontal
363     ## line segments correspond
364     nontrace ← (1:3)[1:3 != tracemar]
365     ## get the dimension names
366     tabnames ← dimnames(outcometab)

```

```

367 ## handle a null desired level setting
368 if (is.null(deslev)) deslev ← 1:length(tabnames[[tracemar]])
369 ## create a palette
370 temPal ← brewer.pal(length(deslev), "Set2")
371 ## calculate the conditional probability distribution of outcome given non-
    trace margins using outcometab
372 condout ← apply(outcometab, nontrace, function(margin) margin/sum(margin))
373 ## and the sums
374 marsums ← apply(outcometab, nontrace, sum)
375 ## extract the desired levels from the margin of interest, write this
    flexibly to allow programmatic margin extraction later
376 ## first define the local environment as the calling environment
377 evEnv ← environment()
378 ## create the language object
379 condoutcall ← quote(condout[,])
380 ## place the levels of interest as the subset argument
381 condoutcall[[tracemar+2]] ← deslev
382 ## evaluate this to subset the data
383 condout ← eval(condoutcall, envir = evEnv)
384 ## rename some useful values for ease of reference
385 ## the dimensionality of the data
386 dims ← dim(condout)
387 ## the number of horizontal segments
388 nseg ← prod(dims)
389 ## the number of 'inner' combination values
390 innern ← dims[nontrace[1]]
391 ## the number of 'outer' combination values (inner and outer refer to axis
    label positions)
392 outern ← dims[nontrace[2]]
393 ## the trace dimension as well
394 tracen ← dims[tracemar]
395 ## add padding between segments to space everything nicely
396 ## first replicate the trace margin sums to create a vector of the correct
    size for the horizontal line generation
397 tempx ← rep(c(marsums)/tracen, times = c(rep(tracen + 1, outern*innern-1),
    tracen))
398 ## replace certain elements with the desired space size, then the cumulative
    sum automatically spaces
399 tempx[(tracen+1)*(1:(innern*outern - 1))] ← space*rep(c(rep(1,innern-1),3),
    length.out = innern*outern-1)*sum(marsums)
400 ## take the cumulative sum to get the positions
401 xpos_line ← c(0, cumsum(tempx)/sum(tempx))
402 ## get the middle positions using the filter function
403 xpos ← c(filter(xpos_line, filter = c(1/2,1/2)))
404 ## remove midsections of padding spaces
405 xpos ← xpos[-(tracen + 1)*(1:(innern*outern - 1))]
406 xpos ← xpos[-length(xpos)]
407 ## use the xpos_line to get widths of sections
408 xposwids ← diff(xpos_line)[-(tracen + 1)*(1:(innern*outern-1))]
409 ## if the expected values are missing, take the original expectation: a
    uniform distribution conditioned on both
410 ## races
411 if (is.null(expected)) expected ← rep(apply(condout, nontrace, mean), each =
    tracen)
412 ## create the empty plot region
413 plot(NA, xlim = range(xpos_line), ylim = c(0, max(condout[,])), xaxt = 'n',
    xlab = "",
414     ylab = "Conditional Probability", ...)
415 ## plot each line and its corresponding expectation
416 invisible(lapply(1:length(xpos), function(ind) {
417     ## plot the horizontal line using the relevant values
418     lines(x = xpos[ind] + xposwids[ind]*c(-1/2,1/2), y = rep(expected[ind],2)
    )
419     ## add the point
420     points(x = xpos[ind], y = condout[ind], col = temPal[((ind - 1) %% tracen
    ) + 1], pch = 19)
421     ## for aesthetics exclude lines if confidence intervals are plotted
422     if (!testlines) lines(x = rep(xpos[ind], 2), y = c(expected[ind], condout
    [ind])),
423                        col = temPal[((ind-1) %% tracen) + 1], lty = 2)

```

```

424   )))
425   ## now add the axes
426   ## first determine axis label positions
427   xpos <- sapply(1:(innern*outern), function(n) mean(xpos[(tracen*(n-1) + 1):(
428     tracen*n])))
429   outpos <- sapply(1:outern, function(n) mean(xpos[(tracen*innern*(n-1) + 1):(
430     tracen*innern*n])))
431   ## add the axis ticks for the inner labels
432   axis(1, at = xpos, labels = rep("", length(xpos)))
433   ## add the labels to the inner axis
434   axis(1, at = xpos, tick = FALSE, labels = rep(tabnames[[nontrace[1]]], times
435     = outern), cex.axis = 0.7,
436     pos = -0.02*max(condout))
437   ## add the labels for the outer axis
438   axis(1, at = outpos, labels = tabnames[[nontrace[2]]], xpd = NA,
439     tick = FALSE, pos = -0.08*max(condout[,]))
440   ## provide the axis title to give context
441   axis(1, at = mean(range(xpos_line)), xpd = NA, tick = FALSE, pos = -0.15*max(
442     condout),
443     labels = paste0("Inner label: ", names(tabnames)[nontrace[1]], " | Outer
444       label: ", names(tabnames)[nontrace[2]]))
445   ## add testing lines if desired
446   if (testlines) {
447     ## get x positions
448     errpos <- xpos
449     ##errpos[3*(1:nseg) - 1] <- xpos
450     ## reformat y positions
451     erry <- c(condout)
452     ## define error bar extensions
453     ext <- c(-0.005, 0.005)
454     ## add bars at each position
455     invisible(sapply(1:length(erry), function(pos) {
456       ## rename the conditional probability for readability
457       p <- erry[pos]
458       ## extract the relevant margin count
459       n <- marsums[floor((pos-1)/dims[tracemar]) + 1]
460       ## calculate the binomial error size
461       err <- 2*sqrt((p/n)*(1-p))
462       ## and the appropriate colour
463       errcol <- temPal[(pos-1) %% dims[tracemar] + 1]
464       ## add vertical lines and horizontal end lines
465       lines(x = errpos[pos] + ext, y = rep(p + err, 2), col = adjustcolor(
466         errcol, alpha.f = 0.5))
467       lines(x = errpos[pos] + ext, y = rep(p - err, 2), col = adjustcolor(
468         errcol, alpha.f = 0.5))
469       lines(x = rep(errpos[pos], 2), y = c(p + err, p - err), col =
470         adjustcolor(errcol, alpha.f = 0.5))
471     })))
472   }
473   ## add a legen to explain the colours
474   legend(x = "top", horiz = TRUE, legend = tabnames[[tracemar]][deslev], col =
475     temPal, inset = -0.04, cex = 0.7,
476     fill = temPal, bg = "white", xpd = NA)
477 }
478
479 ## back to back histogram plot
480 back2backh <- function(data1, data2, cols = NULL, legnames = NULL, ...) {
481   ## get the data1 and data2 names for the legend if no others are provided
482   if (is.null(legnames)) legnames <- c(deparse(substitute(data1)), deparse(
483     substitute(data2)))
484   ## generate and save the histograms of the data
485   hist1 <- hist(data1, plot = FALSE)
486   hist2 <- hist(data2, breaks = hist1$breaks, plot = FALSE)
487   ## use these to generate some necessary plotting parameters
488   maxden <- max(c(hist1$density, hist2$density))
489   nbins <- length(hist1$density)
490   ## generate colours if none are provided
491   if (is.null(cols)) cols <- c("steelblue", "firebrick")
492   ## create an empty plot area
493   plot(NA, xlim = c(-maxden, maxden), ylim = range(hist1$breaks), xlab = "

```

```

    Density", xaxt = 'n', ...)
484 ## add vertical separating line
485 abline(v = 0)
486 ## add an axis
487 axispos ← round(seq(0, maxden, length.out = 3), 2)
488 axis(side = 1, labels = c(axispos[3:2], axispos), at = c(-axispos[3:2],
    axispos))
489 ## plot the histograms back-to-back
490 rect(xleft = -hist1$density, ybottom = hist1$breaks[1:nbins],
491      xright = rep(0, nbins), ytop = hist1$breaks[2:(nbins+1)], col = cols[1])
492 rect(xleft = rep(0, nbins), ybottom = hist2$breaks[1:nbins],
493      xright = hist2$density, ytop = hist2$breaks[2:(nbins+1)], col = cols[2])
494 ## add a legend
495 legend(x = 0, y = max(hist1$breaks), legend = legnames, fill = cols, horiz =
    TRUE, xpd = NA, xjust = 0.5, yjust = 0,
496        bg = "white")
497 }
498
499 ## a function to re-level factor variables to make mosaic plots cleaner (useful
    helper generally)
500 MatRelevel ← function(data) {
501   temp ← lapply(data, function(el) if (is.factor(el)) as.factor(levels(el)[as.
    numeric(el)]) else el)
502   temp ← as.data.frame(temp)
503   names(temp) ← names(data)
504   temp
505 }
506
507 ## a simple helper to convert multiple factor levels into a single 'other' level
508 FactorReduce ← function(vals, tokeep) {
509   chars ← as.character(vals)
510   ## simply replace elements
511   chars[!grepl(paste0(tokeep, collapse = "|"), chars)] ← "Other"
512   chars
513 }
514
515
516 ## DATA INSPECTION #####
517
518 ## load the data
519 if ("FullSunshine_Swapped.csv" %in% list.files(ThesisDir)) {
520   sun.swap ← read.csv(paste0(ThesisDir, "/FullSunshine_Swapped.csv"))
521 } else source(paste0(ThesisDir, "/DataProcess.R"))
522 FullSunshine ← read.csv(paste0(ThesisDir, "/FullSunshine.csv"))
523
524 ## summarize onto the correct scale, the jurors
525 if ("JurorAggregated.Rds" %in% list.files(ThesisDir)) {
526   sun.juror ← readRDS(paste0(ThesisDir, "/JurorAggregated.Rds"))
527 } else sun.juror ← UniqueAgg(sun.swap, by = "JurorNumber", collapse = ",")
528
529 ## also load the data summarized onto the trial scale
530 if ("TrialAggregated.Rds" %in% list.files(ThesisDir)) {
531   sun.trialsum ← readRDS(paste0(ThesisDir, "/TrialAggregated.Rds"))
532 } else warning(paste0("No trial aggregated data found in ", ThesisDir))
533
534 ## there are two juries without charges or other info (noted in the early data
    cleaning but kept for other analysis), remove these
535 sun.trialsum ← sun.trialsum[!(sun.trialsum$TrialNumberID %in% c("590-128", "710-01
    ")),]
536
537 ## display information about juror rejection tendencies
538 mosaicplot(Race ~ Disposition, data = sun.juror, las = 2, shade = TRUE)
539
540 ## create a race filtered data set
541 sun.raceknown ← sun.juror[sun.juror$Race != "U" & sun.juror$DefRace != "U",]
542 sun.raceknown$DefWhiteBlack ← gsub("U", "", sun.raceknown$DefWhiteBlack)
543 sun.raceknown ← MatRelevel(sun.raceknown)
544
545 ## try plotting these
546 mosaicplot(Race ~ PerempStruck, data = sun.raceknown, main = "Race vs. Removal",

```

```

    shade = TRUE, las = 2)
547 mosaicplot(Race ~ Disposition, data = sun.raceknown, main = "Race by Trial Status
    ", shade = TRUE, las = 2)
548 mosaicplot(Race ~ DefStruck, data = sun.raceknown, main = "Race by Defence
    Removal", shade = TRUE, las = 2)
549 mosaicplot(Race ~ ProStruck, data = sun.raceknown, main = "Race by Prosecution
    Removal", shade = TRUE, las = 2)
550 mosaicplot(Race ~ CauseRemoved, data = sun.raceknown, main = "Race by Removal
    with Cause", shade = TRUE, las = 2)
551 ## it seems that there are significantly different strike habits between the
    defense and prosecution, but that
552 ## generally the system does not strike at different rates on average
553 ## recall the paper "Ideological Imbalance and the Peremptory Challenge"
554 par(mfrow = c(1,2))
555 mosaicplot(Race ~ PoliticalAffiliation, data = sun.raceknown[sun.raceknown$Gender
    == "M",],
556     main = "Affiliation and Race (Men)", shade = TRUE, las = 2)
557 mosaicplot(Race ~ PoliticalAffiliation, data = sun.raceknown[sun.raceknown$Gender
    == "F",],
558     main = "Affiliation and Race (Women)", shade = TRUE, las = 2)
559 mosaicplot(Race ~ DefStruck, data = sun.raceknown[sun.raceknown$Gender == "M",],
560     main = "Defense Removals and Race (Men)", shade = TRUE, las = 2)
561 mosaicplot(Race ~ DefStruck, data = sun.raceknown[sun.raceknown$Gender == "F",],
562     main = "Defense Removals and Race (Women)", shade = TRUE, las = 2)
563 mosaicplot(Race ~ ProStruck, data = sun.raceknown[sun.raceknown$Gender == "M",],
564     main = "Prosecution Removals and Race (Men)", shade = TRUE, las = 2)
565 mosaicplot(Race ~ ProStruck, data = sun.raceknown[sun.raceknown$Gender == "F",],
566     main = "Prosecution Removals and Race (Women)", shade = TRUE, las = 2)
567 par(mfrow = c(1,1))
568 ## maybe the same forces are at play here, compare to simulation?
569 ## alternatively, the strong relationship between race and political affiliation
    provides motivation for even an
570 ## unbiased lawyer to preferentially strike one race or the other
571
572 ## these mosaic plots can be confusing, and seemed ineffective upon first
    presentation, try parallel axis plots
573 ## instead
574 ## begin with an overall plot displaying the data at a high level
575 parcoordrace()
576 parcoordracev2(deslev = c(1,2,5))
577 ## but are these differences significant?
578 parcoordracev2(deslev = c(1,2,5), testlines = TRUE)
579
580 ## the independence we want to test here is that of (Race, Disposition)/(
    Defendant Race)
581 ## filter the data to remove small categories
582 sun.chitest <- sun.raceknown
583 sun.chitest$Disposition <- gsub("U_rem", "Unknown", gsub("Foreman", "Kept", sun.
    chitest$Disposition))
584 ## start by generating a table
585 dispTab <- table(sun.chitest[,c("DefWhiteBlack", "Disposition", "WhiteBlack")])
586 ## now apply chi-square tests across the proper margin, start by simply
    generating the residuals
587 dispRes <- lapply(setNames(1:dim(dispTab)[1], dimnames(dispTab)[[1]]), function(
    ind) {
588     ## extract the two way table of this index
589     tab <- dispTab[ind,,]
590     tabdf <- dim(tab) - 1
591     ## calculate the expected values
592     exp <- outer(rowSums(tab), colSums(tab))/sum(tab)
593     ## and residuals
594     resids <- (tab - exp)/sqrt(exp)
595     ## calculate the observed chi-sq value
596     chival <- sum(resids^2)
597     ## and the p value
598     pval <- 1 - pchisq(chival, df = tabdf[1]*tabdf[2])
599     ## return these in a list
600     list(pval = pval, chisq = chival, df = tabdf[1]*tabdf[2], residuals = resids)
601 })
602 ## so, there is a significant difference in behaviour at the 5% level, and it is

```

```

        highly significant for white and black jurors
603
604 ## but these results do not control for much, there could be many factors
        confounding this result
605 ## first create a new data set for the model building
606 sun.jurmod <- sun.raceknown
607 sun.jurmod$DefVisMin <- sun.jurmod$DefWhiteBlack != "White"
608 sun.jurmod$VisMin <- sun.jurmod$WhiteBlack != "White"
609 sun.jurmod$DefStruck <- as.logical(sun.jurmod$DefStruck)
610 sun.jurmod$ProStruck <- as.logical(sun.jurmod$ProStruck)
611 ## now the tricky part, predicting the rejection of a potential juror based on a
        host of factors, the problem is that we must
612 ## perform multinomial regression on the data, but this multinomial regression
        makes comparison of certain parameters
613 ## impossible, i.e. there is no mathematical way to compare the impact of race
        for prosecution and defense rejection statistically
614 ## start by building separate defense and prosecution rejection models
615 mod.def1 <- glm(DefStruck ~ Race + DefRace + Gender + DefGender + CrimeType +
        DefAttyType + PoliticalAffiliation,
616                data = sun.jurmod, family = binomial)
617 ## very poorly fit model, but the reason should be fairly clear, the crime data
        in particular has very specific and small
618 ## classes, try building up the model instead, and using the simpler race
        variable
619 mod.def2 <- glm(DefStruck ~ WhiteBlack*DefWhiteBlack, data = sun.jurmod, family =
        binomial)
620 mod.def3 <- update(mod.def2, formula = DefStruck ~ WhiteBlack + DefWhiteBlack)
621
622 ## idea: instead of multinomial regression, do poisson regression on the dispTab
        above, this allows comparisons
623 sun.rdat <- data.frame(DefRace = rep(c("Black", "Other", "White"), times = 12),
        Disposition = rep(rep(c("C_rem", "D_rem", "Kept", "S_rem"),
624                            , each = 3), times = 3),
625                Race = rep(c("Black", "Other", "White"), each = 12),
626                Count = c(dispTab[,c("C_rem", "D_rem", "Kept", "S_rem"),]))
627 ## estimate the saturated model first
628 mod.rsat <- glm(Count ~ DefRace*Disposition*Race, family = poisson, data = sun.
        rdat)
629 ## now test if the final interaction term can be removed
630 mod.r1 <- update(mod.rsat, formula = Count ~ DefRace*Disposition + DefRace*Race +
        Disposition*Race)
631 ## look at the significance
632 1 - pchisq(mod.r1$deviance, mod.r1$df.residual)
633 ## so, quite clearly, we cannot remove the three way interaction from the model,
        as it is highly significant
634 ## the interpretation: the distribution of strikes, kept, etc. depends on both
        the venire member race and the defendant race
635 ## still, this is perhaps not precise enough, if we change this data to only
        delineate between those kept and the behaviour of
636 ## the lawyers
637 sun.rdat2 <- sun.rdat[sun.rdat$Disposition != "C_rem",]
638 mod.rsat2 <- glm(Count ~ DefRace*Disposition*Race, family = poisson, data = sun.
        rdat2)
639
640 ## this is an interesting result, but perhaps it is related to political
        affiliation (as indicated by the ideological balance
641 ## paper)
642 ## create a table to test this hypothesis
643 dispTab.pol <- table(MatRelevel(sun.chitest[!(sun.chitest$PoliticalAffiliation %in
        % c("Lib", "U"))],
644                        c("Disposition", "PoliticalAffiliation", "WhiteBlack", "DefWhiteBlack"))
        )
645 dispTab.pol <- dispTab.pol[c("C_rem", "D_rem", "Kept", "S_rem"),,,]
646 ## convert to a data frame for fitting
647 sun.pdat <- data.frame(Disp_ = rep(c("C_rem", "D_rem", "Kept", "S_rem"), times =
        27),
648                Pol_ = rep(rep(c("Dem", "Ind", "Rep"), each = 4), times =
        9),
649                Race_ = rep(rep(c("Black", "Other", "White"), each = 12),

```

```

        times = 3),
650         Def_ = rep(c("Black", "Other", "White"), each = 36),
651         Count = c(disptab.pol[,,,]))
652 ## fit a model analogous to those fit above, now controlled for political choices
        in disposition
653 mod.psat <- glm(Count ~ Def_*Disp_*Race_+ Pol_*Disp_, family = poisson, data = sun
        .pdat)
654 ## now remove the third order interaction in the model
655 mod.psatest <- update(mod.psat, formula = Count ~ Def_*Disp_*Race_ + Pol_*Disp_ -
        Def_:Disp_:Race_)
656 ## test the models
657 anova(mod.psat, mod.psatest)
658 1 - pchisq(66.734, 12)
659
660 ## ideological imbalance, look at politics
661 parcoordracev2(table(MatRelevel(sun.raceknown[sun.raceknown$PoliticalAffiliation
        != "U",
662                                c("Disposition",
                                    PoliticalAffiliation",
                                    WhiteBlack")))),
663         deslev = c(1,2,5))
664
665 ## use radial axis plots to view the lawyers tendencies, especially those who act
        as both defence and prosecution lawyers
666 ## maybe remove the top lawyers and remodel
667 ## look at the most prolific lawyers for both sides
668 ## subset the data to only lawyers with one case to remove the lawyer dependency
669
670 ## however, this suggests another question: is this strategy actually successful?
        That is, does there appear to
671 ## be a relation between the number of peremptory challenges and the court case
        outcome?
672 ## this may be difficult, there are a lot of factors to consider:
673 ## - the lawyer and their track record
674 ## - how to judge the success/failure of the case
675 ## see if the presence of challenges is related to the verdict
676 mosaicplot(PerempStruck ~ Guilty, data = sun.swap, main = "Strikes by Guilty",
        shade = TRUE)
677 ## on the level of jurors, this is certainly not the case, but this is not the
        correct scale for the question being
678 ## asked, this question will be addressed again in the case-summarized data
679
680 ## a third obvious question is a comparison of which races strike or keep which
        others, used the synthesized variable
681 ## above to try and identify this
682 mosaicplot(Race ~ StruckBy, data = sun.raceknown, shade = TRUE, main = "Race of
        Juror to Race Removing Juror",
683         las = 2)
684 mosaicplot(Race ~ StruckBy, data = sun.raceknown[sun.raceknown$StruckBy != "Not
        Struck",], shade = TRUE,
685         main = "Race to Race Removing (Only Removed)", las = 2)
686 ## this plot shows no large systematic deviation between the races in their
        rejection habits, this suggests, that
687 ## the rejection that occurs is not as simple as a group identity check
688 ## this might be the wrong race to check, though, perhaps we are better comparing
        the defendant and victim races to
689 ## strike habits
690 par(mfrow = c(1,3))
691 mosaicplot(Race ~ DefRace, data = sun.raceknown[as.logical(sun.raceknown$
        DefStruck),], shade = TRUE,
692         main = "Race of Defense-Struck Jurors to Defendant Race", las = 2)
693 mosaicplot(Race ~ DefRace, data = sun.raceknown[as.logical(sun.raceknown$
        ProStruck),], shade = TRUE,
694         main = "Race of Prosecution-Struck Jurors to Defendant Race", las = 2)
695 mosaicplot(Race ~ DefRace, data = sun.raceknown, las = 2, shade = TRUE, main = "
        Race of Defendant to Venire Race")
696 par(mfrow = c(1,1))
697 ## this makes the defense look as if they are not racist, though the comparison
        to the venire distributions in the third
698 ## panel makes that clearer

```



```

699 ## these distributions to the venire distribution relative to defendant race,
    first combine the smaller races into one
700 ## category to make the plot less noisy and more identifiable
701 ## now look at how the two behave relative in their rejections and their
    acceptance
702 eikos(WhiteBlack ~ DefWhiteBlack + DefStruck, data = sun.raceknown, xlab_rot =
    90,
703     main = "Defense Challenges by Race of Venire Member and Defendant")
704 eikos(WhiteBlack ~ DefWhiteBlack + ProStruck, data = sun.raceknown, xlab_rot =
    90,
705     main = "Prosecution Challenges by Race of Venire Member and Defendant")
706 ## very interesting, the prosecution seems far more aggressive than the defense
707 sun.raceknown$DefWhiteBlack[sun.raceknown$DefWhiteBlack == "Black,U"] <- "Black"
708 sun.raceknown$DefWhiteBlack <- as.factor(as.character(sun.raceknown$DefWhiteBlack)
    )
709 mosaicplot(DefStruck ~ DefWhiteBlack + WhiteBlack, dir = c("v","v","h"), data =
    sun.raceknown, shade = TRUE, las = 2,
710     xlab = "Defendant Race and Defence Removals", ylab = "Juror Race",
    main = "Defence Removal by Defendant Race")
711 mosaicplot(ProStruck ~ DefWhiteBlack + WhiteBlack, dir = c("v","v","h"), data =
    sun.raceknown, shade = TRUE, las = 2,
712     xlab = "Defendant Race and Prosecution Removals", ylab = "Juror Race",
    main = "Prosecution Removal by Defendant Race")
713
714 ## that result is very interesting, the defense strike rates when conditioned on
    defendant race show no racial
715 ## preference, with a preference to reject white jurors regardless of defendant,
    but those of the prosecution do,
716 ## maybe by victim race?
717 mosaicplot(Race ~ VictimRace, data = sun.raceknown[as.logical(sun.raceknown$
    DefStruck)], shade = TRUE,
718     main = "Race of Defense-Struck Jurors to Defendant Race", las = 2)
719 mosaicplot(Race ~ VictimRace, data = sun.raceknown[as.logical(sun.raceknown$
    ProStruck)], shade = TRUE,
720     main = "Race of Prosecution-Struck Jurors to Defendant Race", las = 2)
721 ## hard to see anything there, the majority of victim races are unknown, maybe
    looking at the races removed by defense
722 ## attorney type
723 mosaicplot(DefAttyType ~ Race, data = sun.raceknown[as.logical(sun.raceknown$
    DefStruck)], shade = TRUE, las = 2,
724     main = "Race of Defense-Struck Jurors to Defense Attorney Type")
725 mosaicplot(WhiteBlack ~ DefAttyType, data = sun.raceknown[as.logical(sun.
    raceknown$DefStruck)], shade = TRUE, las = 2,
726     main = "Race of Defense-Struck Jurors to Defense Attorney Type")
727 eikos(WhiteBlack ~ DefWhiteBlack + DefAttyType, data = sun.raceknown[as.logical(
    sun.raceknown$DefStruck)],
728     xlab_rot = 90)
729 ## so what have we seen above is that the prosecuton and defense seem to behave
    very differently in their jury selection
730 ## tactics, the defense seems to reject white individuals at a high rate
    regardless of the defendant, while the prosecution
731 ## seems to prefer the rejection of venire members of the same race as the
    defendant
732
733 ## this last plot shows that different types of lawyers may have different
    strategies, suggests a new investigation:
734 ## that of lawyer strategy and success based on lawyer tendencies, aggregating by
    trial first will be easiest
735
736 ## load the jury summaries
737 if ("AllJuries.Rds" %in% list.files(ThesisDir)) {
738     sun.jursum <- readRDS(paste0(ThesisDir, "/AllJuries.Rds"))
739 } else cat(paste0("No file 'AllJuries.Rds' in ", ThesisDir))
740
741 ## now look at removals across trials for defense and prosecution
742 with(sun.trialsum, plot(jitter(DefRemEst, factor = 2), jitter(ProRemEst, factor =
    2), pch = 20,
743     xlab = "Defense Strike Count (jittered)", ylab = "
    Prosecution Strike Count (jittered)",
744     col = adjustcolor(racePal[as.numeric(DefWhiteBlack)]),

```

```

                                alpha.f = 0.3)))
745 abline(0,1)
746 legend(x = "topleft", legend = levels(sun.trialsum$DefWhiteBlack), col = racePal,
       pch = 20, title = "Defendant Race")
747 ## this is only somewhat informative, it is difficult to see any patterns, use
       the custom posboxplot function
748 ## first encode relative size of point by alpha blending
749 with(sun.trialsum, posboxplot(DefRemEst, ProRemEst, DefWhiteBlack, boxcolours =
       racePal, xlab = "Defense Strike Count",
750                               ylab = "Prosecution Strike Count", boxwidths = 0.8,
                               alphamin = 0.05))
751 ## next by area, another encoding option in this function
752 with(sun.trialsum, posboxplot(DefRemEst, ProRemEst, DefWhiteBlack, boxcolours =
       racePal, xlab = "Defense Strike Count",
753                               ylab = "Prosecution Strike Count", alphaencoding =
                               FALSE, areaencoding = TRUE))
754
755 ## break apart in more detail for the defense
756 DefStruckMeans ← with(sun.trialsum, sapply(levels(DefWhiteBlack),
757                                           function(rc) c(mean((Race.DefRem.
                                           Black/Race.Venire.Black)[
                                           DefWhiteBlack == rc],
758                                                         na.rm = TRUE),
759                                                         mean((Race.DefRem.
                                           White/Race.Venire.
                                           White)[
                                           DefWhiteBlack ==
                                           rc],
                                           na.rm = TRUE))))
760
761 with(sun.trialsum, plot(Race.DefRem.Black/Race.Venire.Black, Race.DefRem.White/
       Race.Venire.White, pch = 20,
762                               xlab = "Black Venire Proportion Struck", ylab = "White
                               Venire Proportion Struck",
763                               xlim = c(0,1), ylim = c(0,1), main = "Defense Strike
                               Proportions",
764                               col = adjustcolor(racePal[as.numeric(DefWhiteBlack)],
                               alpha.f = 0.2)))
765 abline(0,1)
766 points(DefStruckMeans[1,], DefStruckMeans[2,], col = racePal, pch = 4, cex = 2,
       lwd = 1.5)
767 legend(x = "topright", title = "Defendant Race", col = c(racePal,"black"), pch =
       c(rep(20,3),4), bg = "white",
768       legend = c(levels(sun.trialsum$DefWhiteBlack),"Mean"))
769 ## hard to see the patterns at the lines, jitter the proportions
770 with(sun.trialsum, plot(Race.DefRem.Black/Race.Venire.Black + runif(nrow(sun.
       trialsum), min = -0.03, max = 0.03),
771                               Race.DefRem.White/Race.Venire.White, pch = 20,
772                               xlab = "Black Venire Proportion Struck", ylab = "White
                               Venire Proportion Struck",
773                               xlim = c(0,1), ylim = c(0,1), main = "Defense Strike
                               Proportions",
774                               col = adjustcolor(racePal[as.numeric(DefWhiteBlack)],
                               alpha.f = 0.1)))
775
776
777 ## and for the prosecution
778 ProStruckMeans ← with(sun.trialsum, sapply(levels(DefWhiteBlack),
779                                           function(rc) c(mean((Race.ProRem.
                                           Black/Race.Venire.Black)[
                                           DefWhiteBlack == rc],
780                                                         na.rm = TRUE),
781                                                         mean((Race.ProRem.
                                           White/Race.Venire.
                                           White)[
                                           DefWhiteBlack ==
                                           rc],
                                           na.rm = TRUE))))
782
783 with(sun.trialsum, plot(Race.ProRem.Black/Race.Venire.Black, Race.ProRem.White/
       Race.Venire.White, pch = 20,
784                               xlab = "Black Venire Proportion Struck", ylab = "White

```

```

785         Venire Proportion Struck",
        xlim = c(0,1), ylim = c(0,1), main = "Prosecution Strike
        Proportions",
786         col = adjustcolor(racePal[as.numeric(DefWhiteBlack)],
        alpha.f = 0.2)))
787 abline(0,1)
788 points(ProStruckMeans[1,], ProStruckMeans[2,], col = racePal, pch = 4, cex = 2,
        lwd = 1.5)
789 legend(x = "topright", title = "Defendant Race", col = c(racePal,"black"), pch =
        c(rep(20,3),4), bg = "white",
790         legend = c(levels(sun.trialsum$DefWhiteBlack),"Mean"))
791 ## again hard to see, try jittering
792 with(sun.trialsum, plot(Race.ProRem.Black/Race.Venire.Black + runif(nrow(sun.
        trialsum), min = -0.03, max = 0.03),
793         Race.ProRem.White/Race.Venire.White, pch = 20,
794         xlab = "Black Venire Proportion Struck", ylab = "White
        Venire Proportion Struck",
795         xlim = c(0,1), ylim = c(0,1), main = "Prosecution Strike
        Proportions",
796         col = adjustcolor(racePal[as.numeric(DefWhiteBlack)],
        alpha.f = 0.1)))
797
798 ## both of these plots show a much higher proportion of the black venire is
usually struck for both sides, an unsurprising result
799 ## given the the black venire was shown to be smaller in the aggregate statistics
, looking at raw counts next:
800 ## for the defense
801 with(sun.trialsum, plot(jitter(Race.DefRem.Black, factor = 2), jitter(Race.DefRem
        .White, factor = 2), pch = 20,
802         xlab = "Black Venire Strike Count (jittered)", ylab = "
        White Venire Strike Count (jittered)",
803         xlim = c(0,13), ylim = c(0,13), main = "Defense Strike
        Counts",
804         col = adjustcolor(racePal[as.numeric(DefWhiteBlack)],
        alpha.f = 0.2)))
805 legend(x = "topright", title = "Defendant Race", col = racePal, pch = 20, bg = "
        white", legend = levels(sun.trialsum$DefWhiteBlack))
806 ## use custom plot here
807 with(sun.trialsum, posboxplot(Race.DefRem.Black, Race.DefRem.White, DefWhiteBlack
        , racePal,
808         xlab = "Black Venire Strike Count", ylab = "White
        Venire Strike Count",
809         xlim = c(0,13), ylim = c(0,13), main = "Defense
        Strike Counts"))
810 with(sun.trialsum, posboxplot(Race.DefRem.Black, Race.DefRem.White, DefWhiteBlack
        , racePal,
811         xlab = "Black Venire Strike Count", ylab = "White
        Venire Strike Count",
812         xlim = c(0,13), ylim = c(0,13), main = "Defence
        Strike Counts",
813         alphaencoding = FALSE, areaencoding = TRUE))
814
815 ## for the prosecution
816 with(sun.trialsum, plot(jitter(Race.ProRem.Black, factor = 1.2), jitter(Race.
        ProRem.White, factor = 1.2), pch = 20,
817         xlab = "Black Venire Strike Count (jittered)", ylab = "
        White Venire Strike Count (jittered)",
818         xlim = c(0,8), ylim = c(0,8), main = "Prosecution Strike
        Counts",
819         col = adjustcolor(racePal[as.numeric(DefWhiteBlack)],
        alpha.f = 0.2)))
820 legend(x = "topright", title = "Defendant Race", col = racePal, pch = 20, bg = "
        white", legend = levels(sun.trialsum$DefWhiteBlack))
821 ## more of the custom plot
822 with(sun.trialsum, posboxplot(Race.ProRem.Black, Race.ProRem.White, DefWhiteBlack
        , racePal,
823         xlab = "Black Venire Strike Count", ylab = "White
        Venire Strike Count",
824         xlim = c(0,13), ylim = c(0,13), main = "Prosecution
        Strike Counts"))

```

```

825 with(sun.trialsun, posboxplot(Race.ProRem.Black, Race.ProRem.White, DefWhiteBlack
826   , racePal,
827                                     xlab = "Black Venire Strike Count", ylab = "White
                                     Venire Strike Count",
828                                     xlim = c(0,13), ylim = c(0,13), main = "Prosecution
                                     Strike Counts",
829                                     alphaencoding = FALSE, areaencoding = TRUE))
830 ## interesting, this shows some patterns in lawyer behaviour at the trial level
831 ## so there are some obvious patterns we can see in the aggregated data and in
832 the individual cases, see if these affect outcomes
833 with(sun.trialsun, plot(DefRemEst ~ Outcome))
834 with(sun.trialsun, plot(ProRemEst ~ Outcome))
835 ## nothing obvious there, but there is no control for charges/crime type
836 ## compare these to other variables
837 mosaicplot(DefRace ~ CrimeType, data = sun.trialsun, las = 2, main = "Crime and
838   Race", shade = TRUE)
839 mosaicplot(Outcome ~ CrimeType, data = sun.trialsun, las = 2, main = "Crime and
840   Outcome", shade = TRUE)
841 boxplot(DefRemEst ~ CrimeType, data = sun.trialsun)
842 with(sun.trialsun, posboxplot(as.numeric(CrimeType), DefRemEst, DefWhiteBlack,
843   racePal, xaxt = "n",
844                                     ylab = "Defense Strike Count", xlab = "Crime Type")
845   )
846 axis(side = 1, at = 1:7, labels = levels(sun.trialsun$CrimeType))
847 boxplot(ProRemEst ~ CrimeType, data = sun.trialsun)
848 with(sun.trialsun, posboxplot(as.numeric(CrimeType), ProRemEst, DefWhiteBlack,
849   racePal, xaxt = "n",
850                                     ylab = "Prosecution Strike Count", xlab = "Crime
851   Type"))
852 axis(side = 1, at = 1:7, labels = levels(sun.trialsun$CrimeType))
853 ## try using the positional boxplots
854 with(sun.trialsun, posboxplot(DefRemEst, ProRemEst, CrimeType, crimePal))
855 ## too many classes, maybe try drug, sex, theft, other
856 sun.trialsun$DrugSexTheft ← as.factor(FactorReduce(sun.trialsun$CrimeType, tokeep
857   = c("Drug", "Sex", "Theft")))
858 with(sun.trialsun, posboxplot(DefRemEst, ProRemEst, DrugSexTheft, boxcolours =
859   brewer.pal(4, "Set1")))
860 ## also summarize this for the juror data
861 sun.juror$DrugSexTheft ← as.factor(FactorReduce(sun.juror$CrimeType, tokeep = c("
862   Drug", "Sex", "Theft")))
863 ## try something different, plot the tendency of the lawyers themselves
864 ## idea: horizontal axis is lawyers, vertical is strikes
865 LawyerTends ← lapply(unique(c(sun.trialsun$DefAttyName, sun.trialsun$ProsName)),
866   function(name) list(Prosecution = sun.trialsun$ProRemEst[
867     sapply(sun.trialsun$DefAttyName,
868       function(
869         (
870           nms
871         )
872         name
873         %
874         in
875         %
876         nms
877       )
878       ],
879     Defense = sun.trialsun$DefRemEst[sapply(
880       (sun.trialsun$ProsName,
881         function(
882           (
883             nms

```

```

)
name
%
in
%

nms
)
])
)

864 ## order these by those who did both, then defense, then prosecution
865 LawyerOrder <- order(sapply(LawyerTends, function(lst) {
866   lstlens <- sapply(lst, function(el) length(el) > 0)
867   if (all(lstlens)) {
868     0
869   } else if (lstlens[[2]]) {
870     1
871   } else 2}
872 ))
873 ## reorder the lawyer tendencies
874 LawyerTends <- LawyerTends[LawyerOrder]
875 ## plot these
876 plot(NA, xlim = c(1,length(LawyerTends)), ylim = c(0, max(unlist(LawyerTends), na
  .rm = TRUE)))
877 invisible(lapply(1:length(LawyerTends), function(ind) {
878   vals <- LawyerTends[[ind]]
879   points(rep(ind, length(vals$Defense)), vals$Defense, col = adjustcolor("
  steelblue", alpha.f = 0.1), pch = 20)
880   points(rep(ind, length(vals$Prosecution)), vals$Prosecution, col =
  adjustcolor("red", alpha.f = 0.1), pch = 20)
881 })))
882 lines(1:length(LawyerTends), sapply(LawyerTends, function(el) mean(unlist(el), na
  .rm = TRUE)))

```


Epilogue

A few final words.

Declaration of Originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor .

Title of work (in block letters):

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

First name(s):

Muster	Student

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the Citation etiquette information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work .
- I am aware that the work may be screened electronically for plagiarism.
- I have understood and followed the guidelines in the document *Scientific Works in Mathematics*.

Place, date:

Signature(s):

Zurich August 19th 2009	bla

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.