

YALE MEDICAL LIBRARY

AUG 11 2005

T113

+Y12

7241

ABSTRACT

Introduction: Between 1983 and 2005, all 50 states and the District of Columbia have lowered the *per se* blood alcohol concentration (BAC) limit, the BAC considered irrefutable evidence of driving while intoxicated, from 0.10% to 0.08%. Several prior studies have evaluated the effects of the lower limit, with mixed results. Multiple time-series models can be used to study the effects of such policy changes, and can provide the same level of control over confounding factors as randomized trials. The X-11 algorithm is used to control for seasonal and cyclical effects in a time-series. ARIMA is an advanced trend analysis algorithm which can be used to quantify the effects of outside influences on the trends within a time-series.

Methods: We employed time-series regression analysis to evaluate the effects of lowering the *per se* BAC limit to 0.08%. Using the federal Fatality Analysis Reporting System database as a data source, and the monthly number of alcohol-related traffic fatalities as the dependent variable, we conducted this analysis for 31 states and the District of Columbia. We used the X-11 algorithm to control for seasonal variability and the number of weekends in each month. We also controlled for national trends. We used ARIMA analysis to quantify the effect of the law for each state.

Results: Three states had significant decreases in fatalities, and two experienced significant increases, after the passage of 0.08 laws. Overall, 18 states experienced decreases in fatalities, and 14 experienced increases.

Conclusions: While 0.08 laws reduced alcohol-related fatalities in many states, the effect was significant in only a small number of states. Other states experienced increases in fatalities, some of which were significant, however it is unclear what caused these increases.

TABLE OF CONTENTS

Introduction.....	4
Methods.....	11
Results.....	14
Discussion.....	15
Appendix A: Sample SPSS script to create the time-series for one state.....	18
Appendix B: SPSS script to remove the national trend from each state's trend.....	23
Appendix C: Sample SPSS script to calculate the day weights for one state.....	32
Appendix D: Sample SAS script for X11 and ARIMA analysis for one state.....	36
References.....	37

Introduction

Alcohol-related traffic fatalities and the blood alcohol concentration limit

National concern about drunk driving began to increase in the early 1980's after several years of traffic fatality increases. President Reagan formed a national commission on the issue, most states raised the legal drinking age to 21 and established blood alcohol concentration (BAC) limits of 0.10% for impaired driving, and many states raised their penalties for driving while intoxicated (DWI). After years of decreases in alcohol-related crash fatalities, these fatalities as well as total highway fatalities began to increase again in the late 1990's (Table 1). This may be related to the fact that publicity that had accompanied the passage of the new laws was no longer prominent in the media. The increase in fatalities may also be partially explained by one study that found that the increase in enforcement of DWI laws was accompanied by a decrease in enforcement of other traffic laws.¹

During the time period from 1983 to 2005, all 50 states and the District of Columbia (D.C.) have lowered their *per se* BAC limit, the BAC which is considered irrefutable evidence of DWI, from 0.10% to 0.08%. The laws establishing a *per se* BAC limit of 0.08% will hereafter be referred to as 0.08 laws. Several studies have evaluated the effects of these laws, with mixed results.

Table 1: Yearly total traffic fatalities and alcohol-related fatalities since 1982, based on FARS data.

	Total fatalities	Alcohol related fatalities	
Year	Number	Number	Percent
1982	43,945	26,173	60
1983	42,589	24,635	58
1984	44,257	24,762	56
1985	43,825	23,167	53
1986	46,087	25,017	54
1987	46,390	24,094	52
1988	47,087	23,833	51
1989	45,582	22,424	49
1990	44,599	22,587	51
1991	41,508	20,159	49
1992	39,250	18,290	47
1993	40,150	17,908	45
1994	40,716	17,308	43
1995	41,817	17,732	42
1996	42,065	17,749	42
1997	42,013	16,711	40
1998	41,501	16,673	40
1999	41,717	16,572	40
2000	41,945	17,380	41
2001	42,196	17,400	41
2002	43,005	17,524	41
2003	42,643	17,013	40

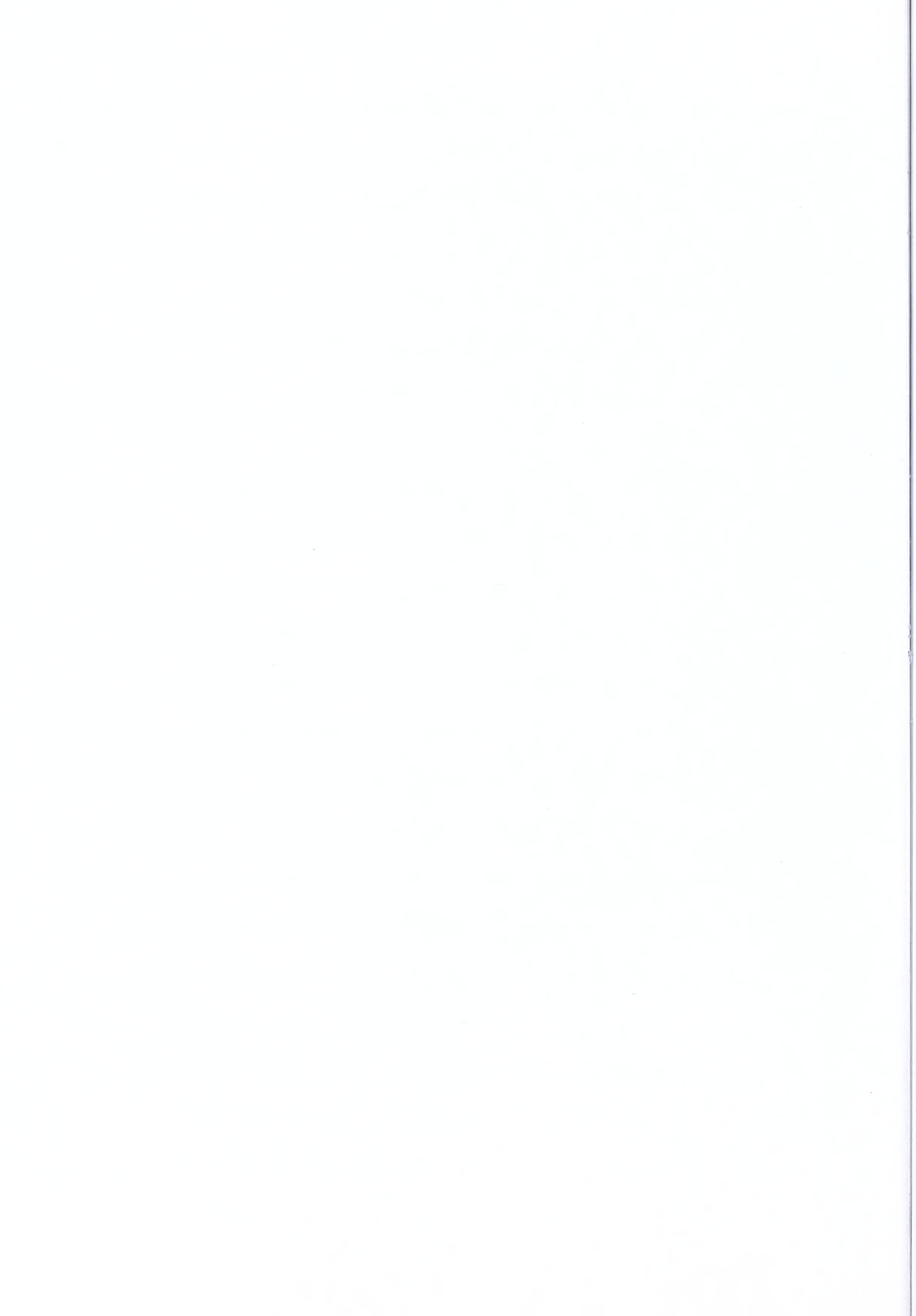
In 1996 Hingson et al. compared each of the first 5 states that passed 0.08 laws with a nearby state which had not passed such a law, and found a statistically significant 16% reduction in alcohol-related fatal crashes in the states that had passed 0.08 laws.² The “Eleven State” study, a 1999 NHTSA study of the first 11 states to pass 0.08 laws, found a decline in alcohol-



related fatalities in eight of the eleven when controlling for ongoing downward trends.³ A 2000 study of alcohol safety laws by Voas et al. used Fatality Analysis Reporting System (FARS) data for all 50 states and controlled for such factors as seasonal changes, unemployment levels, per capita alcohol consumption and annual vehicle miles traveled. Instead of comparing each state with a presumably similar state, this study used a state-by-year matrix for all states as a framework for weighted least squares regression analysis. This study found a significant reduction in alcohol related fatalities associated with 0.08 laws.⁴

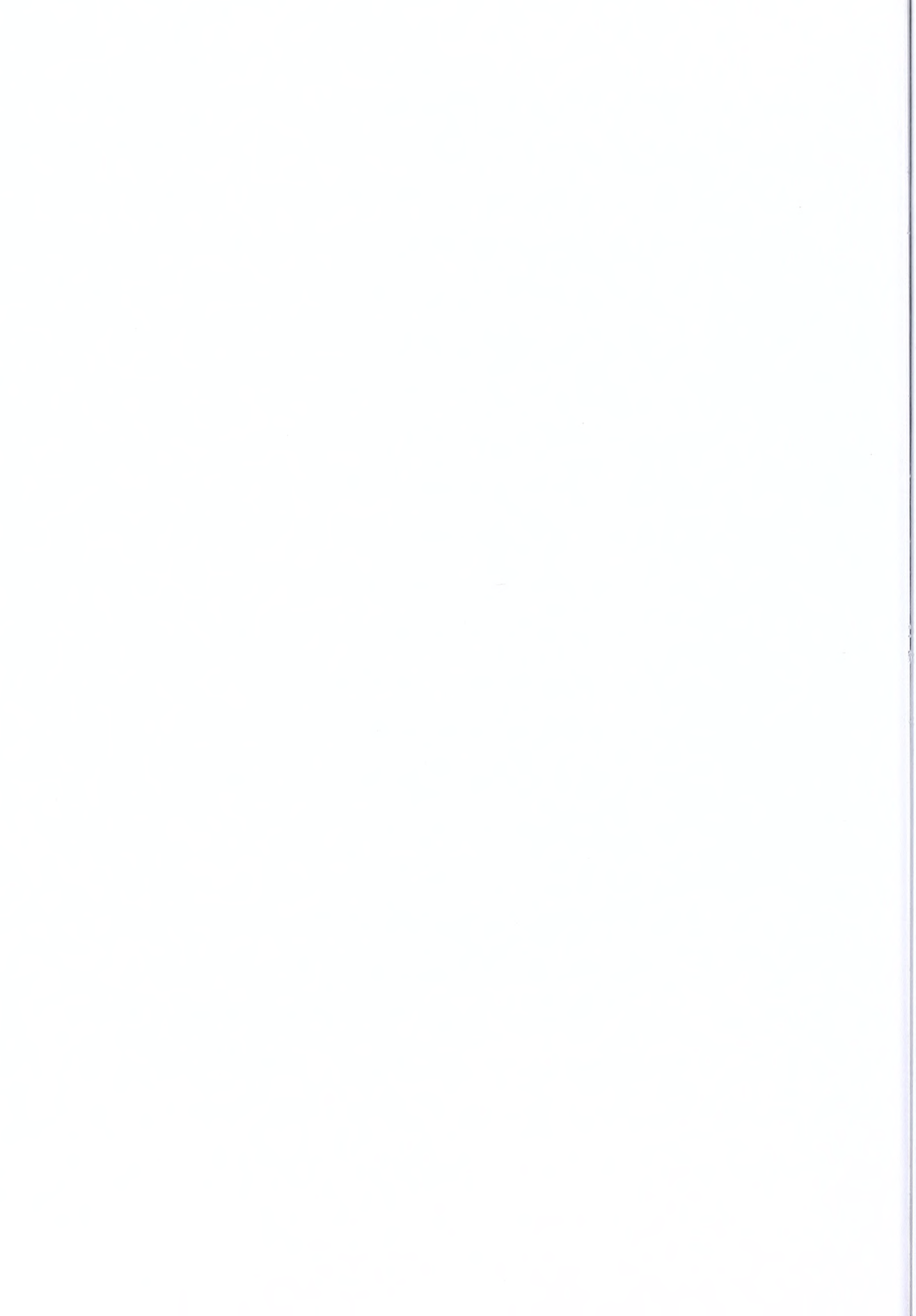
Although these results are promising, there were several weaknesses in these studies. In Hingson et al., the pairing of states has been criticized for not necessarily being based on factors that would influence the number of alcohol-related fatalities. In three of these states Administrative License Revocation (ALR) laws, which are known to reduce alcohol-related fatalities, were also passed within one year of the 0.08 laws, limiting the ability of the study to separate the effects of the two laws. Similarly in the NHTSA “Eleven State” study, 2 of the 8 states with significant findings had enacted ALR laws at around the same time as the 0.08 laws. Additionally, the validity of the regression analysis model used by Voas et al. relies on the assumption that all significant predictors of the outcome variable have been accounted for, and the study did not include measures of media coverage, public attitudes, or any other possibly unknown factors.

Because BAC limits were lowered in many other countries earlier than they were in the United States, international studies are able to show the effects of this policy over longer time periods. Canada and Australia’s lowered BAC limits both were associated with initial



reductions in fatalities, followed by a return to baseline.⁵ A review of other studies of BAC limit reductions internationally (in Britain, Canada and France) has also shown that initial reductions in fatalities, possibly surrounding media coverage of the new laws, eventually dissipated over time.⁶ A study of a lower BAC limit in Denmark found no resulting decrease in alcohol-related crashes.⁷ A similar scenario occurred in Illinois. An initial time-series analysis of the Illinois 0.08 law using FARS data, covering the 18 months after the law went into effect, found a significant reduction in alcohol-related fatalities.⁸ A later study, covering the 24 months after the law went into effect, used two different analysis methods. While a significant reduction in fatalities was still seen using a covariate analysis method, no significant reduction was seen when using a ratio analysis method. The latter method provides greater statistical power by controlling for fluctuations in the drinking-driver time-series which are also present in the nondrinking driver series.⁹ A meta-analysis of drinking-driving laws in the United States also found that the effects of these laws dissipated over time, with the size of effects declining substantially in studies with long follow-up periods. It also found a "publication bias" with journal articles more likely to report successful outcomes of laws, as compared to unpublished studies.¹⁰

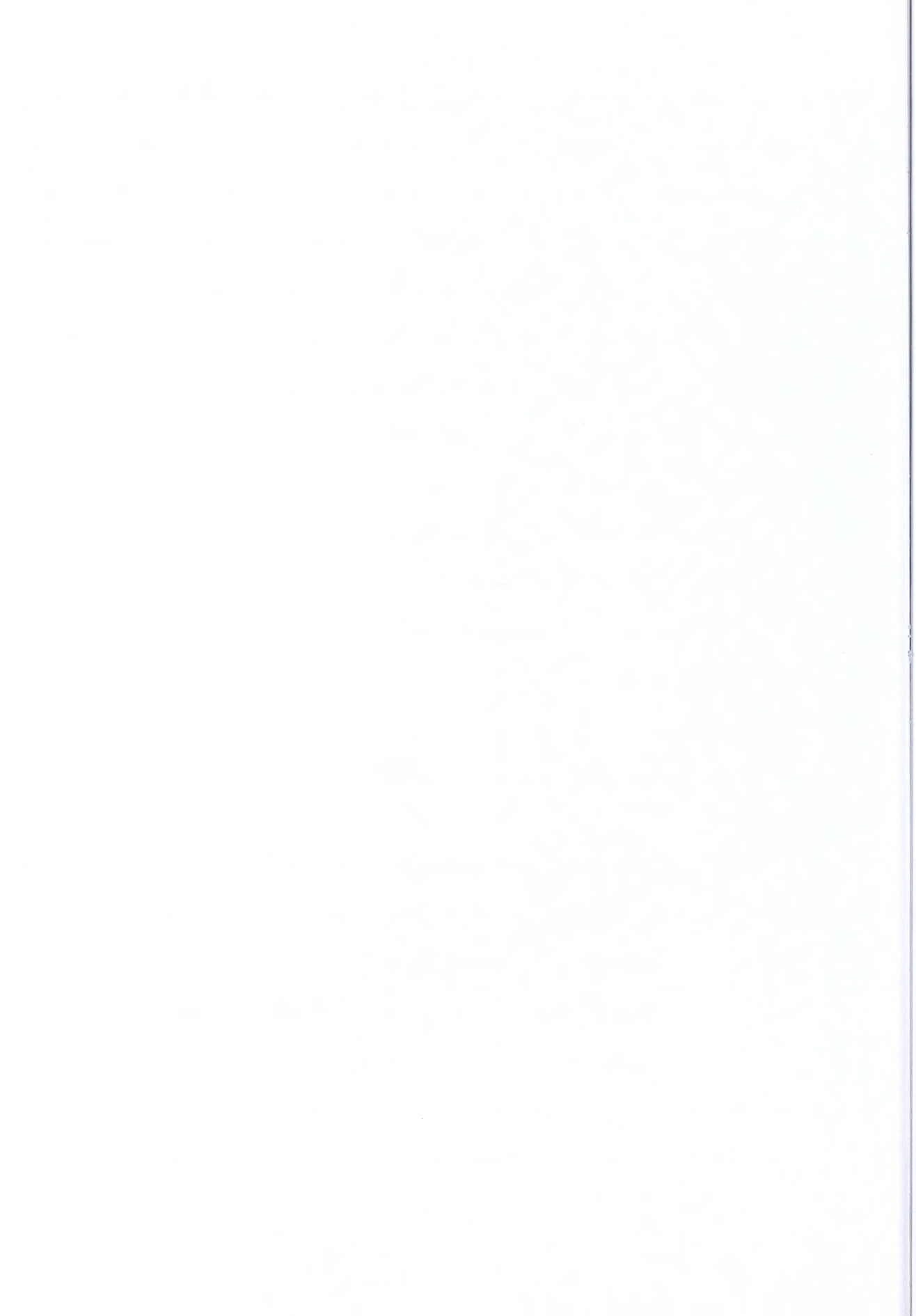
Three other studies of 0.08 laws have used time-series analysis. One study examined the effects of California's 0.08 law, and found no decrease in alcohol-involved crashes or alcohol-related fatalities to be associated with the new law.¹¹ Another study analyzed the effects of the 0.08 law in North Carolina and compared the state to the 37 states that had not changed their BAC limit. Although the study found that the law was well enforced and that a high percentage of alcohol users were aware of the new BAC limit and its implications, the study found no



evidence of a reduction in alcohol-related crashes, or of any increase in the rate of the preexisting downward trend of such crashes.¹² A third study analyzed the effects of Maryland and D.C.'s 0.08 laws, using surrounding states without 0.08 laws to control for influences not related to the law. This study found no significant changes in fatalities in the D.C. area. In Maryland there were no significant decreases in fatalities, but significant increases were found in nighttime crashes, in fatal crashes with drivers with any positive BAC, in crashes with driver BAC at or above 0.08%, and in crashes with driver BAC at or above 0.10%. However, the increase in night crashes relative to day crashes, the increase in drivers with positive BAC relative to drivers with no BAC, and the increase in high-BAC to low-BAC ratios were below the level of significance. Therefore the authors offered no theory to explain the increases in alcohol-related fatalities following the 0.08 law.¹³ Nonetheless, the results of this study indicate the need for further research on the possibility of a link between 0.08 laws and increases in alcohol-related fatalities.

X-11 and ARIMA time-series analysis

It was noted by Campbell and Stanley that multiple time-series models can provide the same control over confounding factors as randomized trials.¹⁴ The X-11 algorithm is a refined version of the Census II method of seasonal adjustment. This algorithm is used to control for seasonal and cyclical effects in a time-series, so that these effects will not interfere with the analysis of other factors impacting the series. ARIMA, which stands for AutoRegressive Integrated Moving Average, is an advanced trend analysis algorithm which can be used to uncover hidden patterns in time-series data, generate forecasts from existing trends, and quantify



the effects of outside influences on the trends within a time-series. Together, the X-11 and ARIMA methods can be used to remove the seasonal components from a time-series, and then to perform analysis on the remaining components.

The X-11 algorithm is an advanced seasonal adjustment method. The concept of seasonal adjustment can be illustrated by the following example. Suppose there was a time-series recording the number of passengers flying each month between the US and Europe over the past 10 years. If this series were plotted, two components would immediately be apparent. First, a general upward trend could be seen, with the number of passengers increasing over the years. Second, a seasonal pattern could be seen within each year, with more travel occurring during the summer and a minor peak occurring in December. Seasonal decomposition is a method of separating these components, by splitting the original series into a trend, a seasonal component, and the remaining variability. The first algorithm to do this is known as the Census I method. This method will separate a time-series into four components, the three mentioned above plus a cyclical component. The cyclical component is similar to the seasonal component, however cyclical factors usually have a duration longer than one year, and the duration of the cycles may vary from one to the next. The Census I method will fit a function to each component which will describe if it is additive, multiplicative, or different type of function. For example, if the amount of air travel increases each summer, it may be found that it increases by roughly the same number of passengers each year over that year's baseline (additive), or that it increases by roughly the same percent over baseline (multiplicative). Likewise an upward trend may increase in a linear fashion (additive) or in an exponential fashion (multiplicative). It is most likely the function that is found to have the best fit for each component will involve a

combination of addition, multiplication, and other operations (e.g. log, etc.). The Census II method, and its variant X-11, add several refinements to Census I. The trading-day adjustment is so named because it was first used to control for the number of days that the financial markets were open during each month. This adjustment allows a weight to be assigned to each day of the week. Therefore it has the flexibility to control not only for trading days (which will have a weight of 1 for Monday through Friday, and 0 for Saturday and Sunday), but also for factors which will have a different weight for each day of the week, for example the daily revenue generated by a restaurant. Census II also allows outliers and extreme values to be removed from a time-series. X-11 allows for successive application of the refinements for outliers, extreme values, and trading-days, in order to successively improve the calculation of the components. The adjusted series which is output by the X-11 algorithm is a time-series with the interference of seasonal and cyclical effects removed, so that the the effects of other influences on the series can be analyzed more accurately.¹⁵

ARIMA is a time-series analysis method developed by Box and Jenkins.¹⁶ This algorithm identifies the mathematical model and parameters which best fit the time-series data. The original application of these models was for forecasting the most likely future values in a time-series. A newer application of the models is known as interrupted time-series analysis. This technique analyzes the change in a time-series at the time of an event which has an influence on the time-series data. A regression analysis algorithm is used to calculate a p-value describing the significance of the influence of the event on the parameters in the model. This is the ideal analysis method for quantifying the effect of a law on a trend which the law is intended to affect.¹⁷

Methods

General approach

We employed time-series regression analysis to evaluate the effects of changing the *per se* BAC limit from 0.10% to 0.08%. Using the federal Fatality Analysis Reporting System (FARS) database as a data source, and the monthly number of alcohol-related traffic fatalities as the dependent variable, we conducted this analysis for each state which had changed the limit and for which we had complete data covering at least 12 months before and after the date of changing the law.¹⁸ We controlled for national trends, seasonal variability, and the number of weekends in each month.

Data source

We used the FARS database as our data source. This database contains one entry for each fatal crash that has occurred during the past 20 years (longer for some states). The entry for each crash includes various information about the crash, as well as 10 imputed driver BAC levels for the crash. These are imputed due to the fact that not every driver in a fatal crash has had his BAC tested or recorded. However, the BAC has been recorded for enough drivers to create a distribution of BACs for each hour of the day, for each year and state. The BAC levels from this distribution are then randomly distributed over the crashes for which the BAC is not known, and this randomization is repeated 10 times to create the 10 imputed BAC values.

Therefore the 10 values may vary widely for an individual crash, however statistics generated from any of the 10 data sets would be accurate and essentially the same.¹⁹ FARS recommends generating statistics from each of the 10 data sets and averaging the results for improved accuracy, and we have done this as part of our method.

Statistical analysis

One time-series was created for each state, with each data point representing the number of alcohol-related fatalities for one month. The number of alcohol-related fatalities was first computed using each of the 10 data sets, by counting the number of fatal crashes with driver BAC $\geq 0.05\%$. The cutoff of 0.05% was selected based on the fact that this is the lowest level at which physiologic alcohol impairment has been demonstrated. The raw value for each month was then calculated by averaging the 10 computed values for greater accuracy.

The adjusted number of alcohol-related fatalities for each month was found by controlling for national trends which could result from national media coverage, improvements in auto safety technology, and changes in driving habits caused by economic factors, gas prices, or other factors. We created a national time-series in which the data point for each month was calculated as the sum of the alcohol-related fatalities in that month in each of the 50 states and D.C. For each state we scaled the national time-series based on the state's population according to the 2000 U.S. Census. The scaled series was subtracted from the raw series, and then a correction factor equal to the mean value of the scaled series was added to each data point to

bring the series back into its original range (this correction factor was necessary due to the fact that our analysis algorithm will not accept negative values). This procedure yielded the adjusted time-series for each state, which we used in our analysis with X-11 and ARIMA.

The X-11 procedure, as implemented in the SAS statistical package, was used to remove seasonal variability and the effect of the number of weekends per month. First, for each state, the number of accidents with BAC $\geq 0.05\%$ occurring on each day of the week was computed for each of the 10 FARS data sets. The 10 values were averaged for each day. The resulting values were input to the X-11 procedure as the “day weights,” thus allowing the procedure to adjust the data points for each month based on the weighted number of days in the month (i.e. allowing the procedure to control for the total number of days in the month as well as the number of weekend days in the month, as weekend days had higher day weight values). The X-11 procedure in SAS was then used on the time-series of each state to remove the effects of both the weighted number of days in each month and the seasonal variability, producing as output a seasonally-adjusted time-series for each state.

ARIMA regression analysis with conditional least squares estimation, as implemented in the SAS package, was used to compute a p-value for the effect of the change to a 0.08% BAC limit. For each state’s seasonally-adjusted time-series, a variable was created which was set to 0 for all months before the BAC limit was lowered, and set to 1 for all months after the limit was lowered. This variable was used as a cross correlation variable in the ARIMA procedure. The ARIMA regression analysis procedure with conditional least squares estimation was used on each state’s seasonally-adjusted time-series. The p-value returned for the effect of the new

variable (set to 0 or 1) in the regression analysis was used as our measure of significance for the effect of the lowered limit for that state.

Results

Table 2 shows the p-value for the effect of 0.08 law in each state analyzed. Three states had significant decreases in fatalities and two experienced significant increases. Overall, 18 states experienced decreases in fatalities after the passage of 0.08 laws, and 14 experienced increases.

Table 2. The effect on fatalities (+/-) and p-value for each state's 0.08 law.

State	Effective date of 0.08% Law	Change in number of fatalities (+/-)	p-value
Alabama	10/1/95	-	0.458
Alaska	9/1/01	-	0.438
Arizona	8/31/01	+	0.721
Arkansas	8/13/01	+	0.550
California	1/1/90	-	0.162
Connecticut	7/1/02	-	0.748
DC	4/13/99	-	0.176
Florida	1/1/94	+	0.224
Georgia	7/1/02	-	0.868
Hawaii	6/30/95	-	0.137
Idaho	7/1/97	-	0.973
Illinois	7/2/97	-	0.551
Indiana	7/1/01	-	0.099
Kansas	7/1/93	+	0.005
Kentucky	10/1/00	-	0.445
Maine	8/4/88	-	0.005
Maryland	9/30/01	+	0.264
Mississippi	7/1/02	-	0.841

Missouri	9/29/01	+	0.559
Nebraska	9/1/01	+	0.097
New Hampshire	1/1/94	-	0.128
New Mexico	1/1/94	-	<0.0001
North Carolina	10/1/93	+	0.296
Oklahoma	7/1/01	+	0.490
Oregon	10/15/83	-	0.999
Rhode Island	7/13/00	+	<0.0001
South Dakota	7/01/02	+	0.297
Utah	8/1/83	+	0.411
Vermont	7/1/91	-	0.045
Virginia	7/1/94	+	0.273
Washington	1/1/99	-	0.318
Wyoming	7/1/02	+	0.292

Discussion

The study presented in this paper is a comprehensive time-series analysis of traffic fatalities after the lowering of the *per se* BAC limit from 0.10% to 0.08% in multiple states. In five states, out of 31 states and D.C., there were significant effects on alcohol-related fatalities associated with the time of implementation of the 0.08 law. Decreases were seen in Maine, New Mexico, and Vermont, while increases occurred in Kansas and Rhode Island.

Assuming the null hypothesis were true, one would expect one to two out of 32 states analyzed to produce a significant result on the basis of chance. Therefore, out of five significant results, at least three are likely to represent the actual effects of the law. Furthermore, with the exception of Vermont, the significant results are strongly significant with $p < 0.01$. Therefore, these results indicate that traffic fatalities have decreased in 18 out of 32 states, with significant

decreases in a small number of states. In addition, 14 out of 32 states experienced increases in fatalities, with a small number of these being significant, but it is unclear what has caused these changes.

The explanations for a significant decrease in fatalities in a small number of states include increased media coverage of the law and of police enforcement efforts, increased fear of arrest and prosecution by drivers, and a resultant decrease in drinking before driving. These explanations and others have been thoroughly explored in the literature.^{2,3,4} Less obvious are the explanations for a significant increase in fatalities in a small number of states. One possible explanation is that 0.08 laws were passed in several states without a great deal of publicity concerning their enactment, or any public awareness campaigns. This may be in part due to the fact that many states enacted these laws because they faced the loss of highway funds if they did not pass the laws. States that had adopted these laws earlier, prior to the federal mandate, were probably more likely to publicize their new law, and focus on DWI enforcement. From this study, it is not clear whether the total amount of time and money devoted to DWI enforcement in most states increased following the implementation of the new law.

There are several limitations to this study. Because FARS is a database of fatal crashes only, we had no data with which to analyze the effect of the law on injuries or on the total number of alcohol-related crashes. For some states not enough time may have passed for the long-term effects of the law to be known, therefore an identical study repeated in several years time would likely shed further light on the effectiveness of the law. Lastly, the actual BAC was

not known for each driver in our data set. While the statistical model ensures that our results are accurate, a study using actual BAC values would be preferable if such data were available.

This study demonstrates that while the 0.08 BAC laws have had their intended effect in many states, the decreases in fatalities have generally not been significant. In addition, a more thorough examination of public awareness and enforcement is necessary before reaching any conclusion as to the overall effectiveness of 0.08 laws. A study that explores the longer term effect of these laws, and takes into account fatality rates per vehicle mile traveled as well as differences in other state policies that may have an impact on DWI fatalities, may serve to answer these questions.

Appendix A: Sample SPSS script to create the time-series for one state.

```
*****
* SPSS Syntax file to create a timeseries for one state.
* First the number of crashes with BAC values (A1 through A10)
* greater than 0.05% (5) is counted for each month for each estimate
* (A1 through A10). Then the estimates are averaged to compute the
* timeseries.
* © 2004 Matthew D. Streckert
*****

* Open the database file for state 01 (Alabama)
GET
  FILE='state01.sav'.
FILTER OFF.
USE ALL.
* Select the cases in which the first BAC estimate is greater than 0.05%
SELECT IF(A1 >= 5).
EXECUTE .
* Sort the cases chronologically
SORT CASES BY YEAR month .
* Count the number of selected cases in each month
CASESTOVARS
  /ID = YEAR month
  /GROUPBY = VARIABLE
  /COUNT = Count .
* Save the resulting estimated timeseries as a .dbf database file
SAVE TRANSLATE OUTFILE='estimate1.dbf'
  /TYPE=DBF /VERSION=2 /MAP /REPLACE.

* Repeat the process for the other BAC estimates, A2-A10

GET
  FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A2 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
  /ID = YEAR month
  /GROUPBY = VARIABLE
  /COUNT = Count2 .
SAVE TRANSLATE OUTFILE='estimate2.dbf'
  /TYPE=DBF /VERSION=2 /MAP /REPLACE.

GET
  FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A3 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
  /ID = YEAR month
  /GROUPBY = VARIABLE
  /COUNT = Count3 .
SAVE TRANSLATE OUTFILE='estimate3.dbf'
  /TYPE=DBF /VERSION=2 /MAP /REPLACE.
```



```
GET
  FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A4 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
  /ID = YEAR month
  /GROUPBY = VARIABLE
  /COUNT = Count4 .
SAVE TRANSLATE OUTFILE='estimate4.dbf'
  /TYPE=DBF /VERSION=2 /MAP /REPLACE.
```

```
GET
  FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A5 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
  /ID = YEAR month
  /GROUPBY = VARIABLE
  /COUNT = Count5 .
SAVE TRANSLATE OUTFILE='estimate5.dbf'
  /TYPE=DBF /VERSION=2 /MAP /REPLACE.
```

```
GET
  FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A6 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
  /ID = YEAR month
  /GROUPBY = VARIABLE
  /COUNT = Count6 .
SAVE TRANSLATE OUTFILE='estimate6.dbf'
  /TYPE=DBF /VERSION=2 /MAP /REPLACE.
```

```
GET
  FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A7 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
  /ID = YEAR month
  /GROUPBY = VARIABLE
  /COUNT = Count7 .
SAVE TRANSLATE OUTFILE='estimate7.dbf'
  /TYPE=DBF /VERSION=2 /MAP /REPLACE.
```

```
GET
  FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A8 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
```



```
/ID = YEAR month
/GROUPBY = VARIABLE
/COUNT = Count8 .
SAVE TRANSLATE OUTFILE='estimate8.dbf'
/TYPE=DBF /VERSION=2 /MAP /REPLACE.
```

```
GET
FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A9 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
/ID = YEAR month
/GROUPBY = VARIABLE
/COUNT = Count9 .
SAVE TRANSLATE OUTFILE='estimate9.dbf'
/TYPE=DBF /VERSION=2 /MAP /REPLACE.
```

```
GET
FILE='state01.sav'.
FILTER OFF.
USE ALL.
SELECT IF(A10 >= 5).
EXECUTE .
SORT CASES BY YEAR month .
CASESTOVARS
/ID = YEAR month
/GROUPBY = VARIABLE
/COUNT = Count10 .
SAVE TRANSLATE OUTFILE='estimate10.dbf'
/TYPE=DBF /VERSION=2 /MAP /REPLACE.
```

* Translate the .dbf files to SPSS native format

```
GET TRANSLATE
FILE='estimate1.dbf'
/TYPE=DBF /MAP .
SAVE OUTFILE='count1.sav'
/COMPRESSED.
```

```
GET TRANSLATE
FILE='estimate2.dbf'
/TYPE=DBF /MAP .
SAVE OUTFILE='count2.sav'
/COMPRESSED.
```

```
GET TRANSLATE
FILE='estimate3.dbf'
/TYPE=DBF /MAP .
SAVE OUTFILE='count3.sav'
/COMPRESSED.
```

```
GET TRANSLATE
FILE='estimate4.dbf'
/TYPE=DBF /MAP .
SAVE OUTFILE='count4.sav'
/COMPRESSED.
```



```
GET TRANSLATE
  FILE='estimate5.dbf'
  /TYPE=DBF /MAP .
SAVE OUTFILE='count5.sav'
/COMPRESSED.
```

```
GET TRANSLATE
  FILE='estimate6.dbf'
  /TYPE=DBF /MAP .
SAVE OUTFILE='count6.sav'
/COMPRESSED.
```

```
GET TRANSLATE
  FILE='estimate7.dbf'
  /TYPE=DBF /MAP .
SAVE OUTFILE='count7.sav'
/COMPRESSED.
```

```
GET TRANSLATE
  FILE='estimate8.dbf'
  /TYPE=DBF /MAP .
SAVE OUTFILE='count8.sav'
/COMPRESSED.
```

```
GET TRANSLATE
  FILE='estimate9.dbf'
  /TYPE=DBF /MAP .
SAVE OUTFILE='count9.sav'
/COMPRESSED.
```

```
GET TRANSLATE
  FILE='estimate10.dbf'
  /TYPE=DBF /MAP .
SAVE OUTFILE='count10.sav'
/COMPRESSED.
```

```
* Open a blank timeseries
GET
  FILE='blankseries.sav'.
```

* Add the data from each estimate in a separate column

```
MATCH FILES /FILE=*
  /FILE='count1.sav'
  /BY year month.
EXECUTE.
```

```
MATCH FILES /FILE=*
  /FILE='count2.sav'
  /BY year month.
EXECUTE.
```

```
MATCH FILES /FILE=*
  /FILE='count3.sav'
  /BY year month.
EXECUTE.
MATCH FILES /FILE=*
```



```

/FILE='count4.sav'
/BY year month.
EXECUTE.
MATCH FILES /FILE=*
/FILE='count5.sav'
/BY year month.
EXECUTE.
MATCH FILES /FILE=*
/FILE='count6.sav'
/BY year month.
EXECUTE.
MATCH FILES /FILE=*
/FILE='count7.sav'
/BY year month.
EXECUTE.
MATCH FILES /FILE=*
/FILE='count8.sav'
/BY year month.
EXECUTE.
MATCH FILES /FILE=*
/FILE='count9.sav'
/BY year month.
EXECUTE.
MATCH FILES /FILE=*
/FILE='count10.sav'
/BY year month.
EXECUTE.

```

```

* If the count variable for any month is missing, this is
* because zero events were counted. Therefore create
* the proper variable and set its value to zero.

```

```

IF (missing(count)=1) count = 0 .
EXECUTE .
IF (missing(count2)=1) count2 = 0 .
EXECUTE .
IF (missing(count3)=1) count3 = 0 .
EXECUTE .
IF (missing(count4)=1) count4 = 0 .
EXECUTE .
IF (missing(count5)=1) count5 = 0 .
EXECUTE .
IF (missing(count6)=1) count6 = 0 .
EXECUTE .
IF (missing(count7)=1) count7 = 0 .
EXECUTE .
IF (missing(count8)=1) count8 = 0 .
EXECUTE .
IF (missing(count9)=1) count9 = 0 .
EXECUTE .
IF (missing(count10)=1) count10 = 0 .
EXECUTE .

```

```

* Average the ten timeseries into one final timeseries

```

```

COMPUTE AVGCOUNT = (count + count2 + count3 + count4 + count5 + count6 + count7 +
count8 + count9 + count10) / 10 .
EXECUTE .

```

```

* Save the results in a file

```

```

SAVE TRANSLATE OUTFILE='avgseries01.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.

```


Appendix B: SPSS script to remove the national trend from each state's trend

```
*****
* SPSS syntax to remove the national trend from each state's trend.
* First the national series is adjusted to have an average value
* of zero, so that it will not change the range of each state's
* series when the adjustment is done. This is done by subtracting
* 2222 (the average value) from each data point in the national
* series. Then the national series, for each state, is adjusted for
* that state's population. This is done by multiplying by the
* state's population over the national population. Lastly the final
* timeseries is saved for each state.
* © 2004 Matthew D. Streckert
*****
```

```
*****Alabama
```

```
* Open the timeseries for state 01 (Alabama)
```

```
GET TRANSLATE
  FILE='avgseries01.dbf'
  /TYPE=DBF /MAP .
```

```
* Add the data from the national timeseries in a new column
```

```
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
```

```
* Calculate the adjusted count for each month as described above
```

```
COMPUTE adjusted = avgcount - ((national-2222) * (4478896/287973924)) .
EXECUTE .
```

```
* Save the final timeseries for state 01 (Alabama)
```

```
SAVE TRANSLATE OUTFILE='finalts01.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

```
*****Alaska
```

```
GET TRANSLATE
```

```
  FILE='avgseries02.dbf'
  /TYPE=DBF /MAP .
```

```
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
```

```
COMPUTE adjusted = avgcount - ((national-2222) * (641482/287973924)) .
EXECUTE .
```

```
SAVE TRANSLATE OUTFILE='finalts02.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

```
*****Arizona
```

```
GET TRANSLATE
```

```
  FILE='avgseries04.dbf'
  /TYPE=DBF /MAP .
```

```
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
```



```
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (5441125/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts04.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Arkansas

```
GET TRANSLATE
  FILE='avgseries05.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (2706268/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts05.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****California

```
GET TRANSLATE
  FILE='avgseries06.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (35001986/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts06.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Connecticut

```
GET TRANSLATE
  FILE='avgseries09.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (3458587/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts09.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****DC

```
GET TRANSLATE
  FILE='avgseries11.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
```



```
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (569157/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts11.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Florida

```
GET TRANSLATE
  FILE='avgseries12.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (16691701/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts12.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Georgia

```
GET TRANSLATE
  FILE='avgseries13.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (8544005/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts13.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Hawaii

```
GET TRANSLATE
  FILE='avgseries15.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (1240663/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts15.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Idaho

```
GET TRANSLATE
  FILE='avgseries16.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
```



```
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (1343124/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts16.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Illinois

```
GET TRANSLATE
  FILE='avgseries17.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (12586447/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts17.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Indiana

```
GET TRANSLATE
  FILE='avgseries18.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (6156913/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts18.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Kansas

```
GET TRANSLATE
  FILE='avgseries20.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (2711769/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts20.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Kentucky

```
GET TRANSLATE
  FILE='avgseries21.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
```



```
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (4089822/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts21.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Louisiana

```
GET TRANSLATE
FILE='avgseries22.dbf'
/TYPE=DBF /MAP .
MATCH FILES /FILE=*
/FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (4476192/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts22.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Maine

```
GET TRANSLATE
FILE='avgseries23.dbf'
/TYPE=DBF /MAP .
MATCH FILES /FILE=*
/FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (1294894/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts23.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Maryland

```
GET TRANSLATE
FILE='avgseries24.dbf'
/TYPE=DBF /MAP .
MATCH FILES /FILE=*
/FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (5450525/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts24.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Mississippi

```
GET TRANSLATE
FILE='avgseries28.dbf'
/TYPE=DBF /MAP .
MATCH FILES /FILE=*
/FILE='national-ts.sav'.
```



```
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (2866733/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts28.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Missouri

```
GET TRANSLATE
FILE='avgseries29.dbf'
/TYPE=DBF /MAP .
MATCH FILES /FILE=*
/FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (5669544/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts29.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Nebraska

```
GET TRANSLATE
FILE='avgseries31.dbf'
/TYPE=DBF /MAP .
MATCH FILES /FILE=*
/FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (1727564/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts31.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****New Hampshire

```
GET TRANSLATE
FILE='avgseries33.dbf'
/TYPE=DBF /MAP .
MATCH FILES /FILE=*
/FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (1274405/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts33.dbf'
/TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****New Mexico

```
GET TRANSLATE
FILE='avgseries35.dbf'
/TYPE=DBF /MAP .
MATCH FILES /FILE=*
/FILE='national-ts.sav'.
```



```
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (1852044/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts35.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****North Carolina

```
GET TRANSLATE
  FILE='avgseries37.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (8305820/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts37.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Oklahoma

```
GET TRANSLATE
  FILE='avgseries40.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (3489700/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts40.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Oregon

```
GET TRANSLATE
  FILE='avgseries41.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (3520355/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts41.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Rhode Island

```
GET TRANSLATE
  FILE='avgseries44.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
```



```
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (1068326/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts44.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****South Dakota

```
GET TRANSLATE
  FILE='avgseries46.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (760437/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts46.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Tennessee

```
GET TRANSLATE
  FILE='avgseries47.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (5789796/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts47.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Texas

```
GET TRANSLATE
  FILE='avgseries48.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (21736925/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts48.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Utah

```
GET TRANSLATE
  FILE='avgseries49.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (2318789/287973924)) .
```



```
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts49.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Vermont

```
GET TRANSLATE
  FILE='avgseries50.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (616408/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts50.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Virginia

```
GET TRANSLATE
  FILE='avgseries51.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (7287829/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts51.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Washington

```
GET TRANSLATE
  FILE='avgseries53.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (6067060/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts53.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```

*****Wyoming

```
GET TRANSLATE
  FILE='avgseries56.dbf'
  /TYPE=DBF /MAP .
MATCH FILES /FILE=*
  /FILE='national-ts.sav'.
EXECUTE.
COMPUTE adjusted = avgcount - ((national-2222) * (498830/287973924)) .
EXECUTE .
SAVE TRANSLATE OUTFILE='finalts56.dbf'
  /TYPE=DBF /VERSION=3 /MAP /REPLACE.
```


Appendix C: Sample SPSS script to calculate the day weights for one state (Connecticut)

```
*****
* SPSS syntax to calculate a state's day weights.  The number
* of alcohol-related crashes in the database for each day of
* the week is calculated for each of the 10 sets of imputed
* values.  The ten sets of day weights are then averaged.
* © 2004 Matthew D. Streckert
*****

* Open the database for state 09 (Connecticut)
GET
  FILE='state09.sav'.

* Keep only those records in which the first imputed BAC is >= 0.05
FILTER OFF.
USE ALL.
SELECT IF(a1 >= 5).
EXECUTE .

* Convert the date into a format that SPSS understands
COMPUTE VAR00001 = DATE.MDY(month,day, YEAR) .
VARIABLE LABELS VAR00001 'date' .
EXECUTE .

* Use the date to compute the day of the week for each record
COMPUTE DayOfWeek = XDATE.WKDAY(VAR00001) .
EXECUTE .

* Save the records in an intermediate file
SAVE OUTFILE='days1.sav'
  /COMPRESSED.

* Count the number of records for each day of the week in the file
MATCH FILES /FILE=*
  /RENAME (VAR00001 = d0)
  /FILE='days1.sav'
  /KEEP= DayOfWeek.
EXECUTE.
SORT CASES BY DayOfWeek .
CASESTOVARS
  /ID = DayOfWeek
  /GROUPBY = VARIABLE
  /COUNT = Weights1 .

* Save the day weights in a file
SAVE OUTFILE='weights1.sav'
  /COMPRESSED.

* Repeat the same process for imputed values 2-10

GET
  FILE='state09.sav'.

FILTER OFF.
USE ALL.
SELECT IF(a2 >= 5).
```



```
EXECUTE .
COMPUTE VAR00001 = DATE.MDY(month,day,YEAR) .
VARIABLE LABELS VAR00001 'date' .
EXECUTE .
```

```
COMPUTE DayOfWeek = XDATE.WKDAY(VAR00001) .
EXECUTE .
```

```
SAVE OUTFILE='days2.sav'
/COMPRESSED.
```

```
MATCH FILES /FILE=*
/RENAME (VAR00001 = d0)
/FILE='days2.sav'
/KEEP= DayOfWeek.
EXECUTE.
```

```
SORT CASES BY DayOfWeek .
CASESTOVARS
/ID = DayOfWeek
/GROUPBY = VARIABLE
/COUNT = Weights2 .
SAVE OUTFILE='weights2.sav'
/COMPRESSED.
```

```
GET
FILE='state09.sav'.
```

```
FILTER OFF.
USE ALL.
SELECT IF(a9 >= 5) .
EXECUTE .
COMPUTE VAR00001 = DATE.MDY(month,day,YEAR) .
VARIABLE LABELS VAR00001 'date' .
EXECUTE .
```

```
COMPUTE DayOfWeek = XDATE.WKDAY(VAR00001) .
EXECUTE .
```

```
SAVE OUTFILE='days9.sav'
/COMPRESSED.
```

```
MATCH FILES /FILE=*
/RENAME (VAR00001 = d0)
/FILE='days9.sav'
/KEEP= DayOfWeek.
EXECUTE.
```

```
SORT CASES BY DayOfWeek .
```



```
CASESTOVARS
  /ID = DayOfWeek
  /GROUPBY = VARIABLE
  /COUNT = Weights9 .
SAVE OUTFILE='weights9.sav'
  /COMPRESSED.
```

```
GET
  FILE='state09.sav'.
```

```
FILTER OFF.
USE ALL.
SELECT IF(a10 >= 5).
EXECUTE .
COMPUTE VAR00001 = DATE.MDY(month,day,YEAR) .
VARIABLE LABELS VAR00001 'date' .
EXECUTE .
```

```
COMPUTE DayOfWeek = XDATE.WKDAY(VAR00001) .
EXECUTE .
```

```
SAVE OUTFILE='days10.sav'
  /COMPRESSED.
```

```
MATCH FILES /FILE=*
  /RENAME (VAR00001 = d0)
  /FILE='days10.sav'
  /KEEP= DayOfWeek.
EXECUTE.
```

```
SORT CASES BY DayOfWeek .
CASESTOVARS
  /ID = DayOfWeek
  /GROUPBY = VARIABLE
  /COUNT = Weights10 .
SAVE OUTFILE='weights10.sav'
  /COMPRESSED.
```

* Combine the 10 saved files into one file

```
GET
  FILE='weights1.sav'.
MATCH FILES /FILE=*
  /FILE='weights2.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.
MATCH FILES /FILE=*
  /FILE='weights3.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.
MATCH FILES /FILE=*
  /FILE='weights4.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.
```



```

MATCH FILES /FILE=*
  /FILE='weights5.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.
MATCH FILES /FILE=*
  /FILE='weights6.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.
MATCH FILES /FILE=*
  /FILE='weights7.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.
MATCH FILES /FILE=*
  /FILE='weights8.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.
MATCH FILES /FILE=*
  /FILE='weights9.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.
MATCH FILES /FILE=*
  /FILE='weights10.sav'
  /RENAME (DayOfWeek = d0)
  /DROP= d0.
EXECUTE.

```

* Average the 10 sets of day weights

```

COMPUTE PDWEIGHTS = (Weights1 + Weights2 + Weights3 + Weights4 + Weights5 + Weights6 +
Weights7 + Weights8 + Weights9 + Weights10) / 10 .
EXECUTE .

```

* Save the day weights for state 09

```

SAVE OUTFILE='pdweights09.sav'
  /COMPRESSED.

```


Appendix D: Sample SAS script for X11 and ARIMA analysis for one state (Alabama).

```
#####
# SAS script to use X-11 and ARIMA to analyze the
# timeseries for one state.
# © 2004 Matthew D. Streckert
#####

# Run the X-11 procedure on the timeseries for state 01 (Alabama)
proc x11 data=finalts01;

    monthly date=date tdregr=adjust;

    # Use the day weights calculated for this state

    pdweights sun=1.6623 mon=0.7423 tue=0.7078 wed=0.736 thu=0.896 fri=1.4855
    sat=2.4063 ;

    var adjusted;

    # Output a series called x11out

    output out=x11out d11=Finalseasadjser;

run;

data newdata;

    set x11out;
    # Add a variable law which is set to one for
    # dates when the law was in effect; otherwise
    # it set to zero.
    law = date >= '1oct1995'd;

run;

# Run the ARIMA procedure with law as the cross-correlation variable.
proc arima data=newdata;
    identify var=Finalseasadjser crosscorr=law;
    estimate p=1 q=1 input=law;
run;
```


- ¹ Hingson R, Howland J, et al. Effects of Legal Penalty Changes and Laws to Increase Drunken Driving Convictions on Fatal Traffic Crashes. *Bulletin of New York Academic Medicine*. 64(7), 1988.
- ² Hingson R, Heeren T, Winter M. Lowering State Legal Blood Alcohol Limits to 0.08%: The Effect on Fatal Motor Vehicle Crashes. *American Journal of Public Health*. 86(9), 1996.
- ³ Aspler R, Char AR, Harding WM, Klein TM. The Effects of 0.08 BAC Laws. NHTSA April 1999.
- ⁴ Voas RB, Tippetts AS, Fell JC. The Relationship of Alcohol Safety Laws to Drinking Drivers in Fatal Crashes. *Accident Analysis and Prevention*. 32(4), 2000.
- ⁵ Mann RE, Macdonald S, et al. The Effects of Introducing or Lowering Legal per se Blood Alcohol Limits for Driving: an International Review. *Accident Analysis and Prevention*. 33, 2001.
- ⁶ Ross H. *Deterring the Drinking Driver: Legal Policy and Social Control*. Lexington Books. 1982.
- ⁷ Bernhoft IM, Behrendorff. Effect of Lowering the Alcohol Limit in Denmark. *Accident Analysis and Prevention*. 35(4), 2003.
- ⁸ Voas RB, Taylor E, Baker T, Tippetts AS. Effectiveness of the Illinois .08 Law. NHTSA Dec 2000.
- ⁹ Voas RB, Tippetts AS, Taylor E. Evaluation of the Illinois .08 Law: An Update with the 1999 FARS Data. NHTSA Dec 2001.
- ¹⁰ Wagenaar AC, Zobeck TS, Williams GD, Hingson R. Methods Used in Studies of Drink-Drive Control Efforts: a Meta-Analysis of the Literature from 1960 to 1991. *Accident Analysis And Prevention*. 27(3), 1995.
- ¹¹ Rogers PN, Schoenig SE. A Time Series Evaluation of California's 1982 DUI Legislative Reforms. *Accident Analysis and Prevention*. 26, 1994.
- ¹² Foss RD, Stewart JR, Reinfurt DW. Evaluation of the Effects of North Carolina's 0.08% BAC Law. *Accident Analysis and Prevention*. 33, 2001.
- ¹³ Zador PL, Rauch WJ, Ahlin EM, Duncan D. Effects of the .08 g/dL BAC Law on Fatal Crashes in the State of Maryland, USA. Unpublished. 2004.
- ¹⁴ Campbell DT, Stanley JC. *Experimental and quasi-experimental designs for research*. Houghton Mifflin. 1963.
- ¹⁵ Statsoft Inc. *Electronic Statistics Textbook*. WEB: <http://www.statsoft.com/textbook/stathome.html> Statsoft Inc., Tulsa, OK. 2004.
- ¹⁶ Box GP, Jenkins GM. *Timeseries Analysis: Forecasting and Control*. Holden-Day, San Francisco. 1976.
- ¹⁷ SAS Institute Inc. *SAS/ETS User's Guide, Version 6, Second Edition*. SAS Institute, Cary, NC. 1993.
- ¹⁸ NHTSA. FARS Database. WEB: <http://www-fars.nhtsa.dot.gov>
- ¹⁹ Subramanian R. Transitioning to Multiple Imputation – A New Method to Estimate Missing Blood Alcohol Concentration (BAC) Values in FARS. NHTSA 2002.



3 9002 07949 1578

**HARVEY CUSHING/JOHN HAY WHITNEY
MEDICAL LIBRARY**

MANUSCRIPT THESES

Unpublished theses submitted for the Master's and Doctor's degrees and deposited in the Medical Library are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but passages must not be copied without permission of the authors, and without proper credit being given in subsequent written or published work.

This thesis by
has been used by the following person, whose signatures attest their acceptance of the above restrictions.

NAME AND ADDRESS

DATE

