# US Traffic Accidents: Statistical Methods for Discrete Response, Time Series, and Panel Data in R

Prathyusha Charagondla, Chris Weyandt and Andrew Kiruluta

## U.S. traffic fatalities: 1980-2004

## 1. Exploratory Data Analysis:

This data is structured as a long panel set, where each of the 48 continental states are numbered alphabetically from 1 to 51, with 2, 9, and 12 missing (Alaska, Hawaii, District of Columbia). Each state has associated with it 25 observations ranging from 1980 to 2004. The year is indicated both as its own variable and represented as one of 25 dummy variables. Within each Year-by-State observation there are observations that describe the state's traffic laws, traffic Fatalities, and population demographics.

Variables that are coded as dummy (1 or 0) will often show a number between these two dichotomous options. This indicates that the state's law was changed during this year and the fraction indicates the portion of the year for which the variable was active. For example, a value of 0.75 indicates that for three quarters of the year, the variable was $True$ and for the other quarter, the variable was $False$.

```
driving <- load("driving.RData")
data.desc <- describe(data)
r.vars <- c("totfatrte")
e.vars <- c("bac08", "bac10", "perse", "sbprim", "sbsecon", "sl70plus", "gdl", "perc14_24",
    "unem", "vehicmilespc")
p.vars <- c("year", e.vars)
d.vars <- grep("d\\d\\d", names(data), value = TRUE)
m.vars <- c("year", "state")
t.vars <- c("sl70", "sl75", "slnone")
u.vars <- (names(data) %>%
    setdiff(r.vars) %>%
    setdiff(e.vars) %>%
    setdiff(d.vars) %>%
    setdiff(m.vars) %>%
    setdiff(t.vars))
a.vars <- names(data) %>%
    setdiff(u.vars)
```

Each attribute should have 1200 values (48 states, 25 years). We can see that we have no missing values. The only immediate concern here is that some boolean legislature indicators (bac08, bac10, each of the sl variables) have more than 2 values. This is due to the fact that some of the laws

were implemented mid-year - thus, the fraction indicates which month of the year the transition ocurred.

```
summaryAttributes <- do.call(bind_rows, c(list(.id = "variable"), lapply(data.desc, function(co
    as.numeric)))) %>%
    inner_join(desc %>%
        mutate_if(is.factor, as.character), y = .) %>%
    select(-matches("\\.\\d\\d")) %>%
    filter(variable %in% setdiff(a.vars, d.vars))
```

```
## Joining, by = "variable"
```

```
summaryAttributes[c("variable", "label", "missing", "distinct", "Info", "Mean", "Gmd")]
```

```
##        variable                                          label missing distinct
## 1          year                                1980 through 2004       0       25
## 2         state           48 continental states, alphabetical       0       48
## 3          sl70                                speed limit == 70       0       14
## 4          sl75                                speed limit == 75       0        9
## 5        slnone                                   no speed limit       0        3
## 6           gdl                graduated drivers license law       0        8
## 7         bac10                      blood alcohol limit .10       0       10
## 8         bac08                      blood alcohol limit .08       0        8
## 9         perse administrative license revocation (per se law)       0        9
## 10    totfatrte       total fatalities per 100,000 population       0      916
## 11         unem                 unemployment rate, percent       0      112
## 12    perc14_24       percent population aged 14 through 24       0       87
## 13     sl70plus                          sl70 + sl75 + slnone       0       15
## 14       sbprim                =1 if primary seatbelt law       0        2
## 15      sbsecon               =1 if secondary seatbelt law       0        2
## 16 vehicmilespc                                                  0     1200
##     Info      Mean       Gmd
## 1  0.998 1.992e+03 8.327e+00
## 2  1.000 2.715e+01 1.660e+01
## 3  0.333 1.190e-01 2.098e-01
## 4  0.231 8.024e-02 1.477e-01
## 5  0.025 7.569e-03 1.504e-02
## 6  0.449 1.741e-01 2.877e-01
## 7  0.748 6.231e-01 4.691e-01
## 8  0.540 2.135e-01 3.358e-01
## 9  0.760 5.471e-01 4.958e-01
## 10 1.000 1.892e+01 7.032e+00
## 11 1.000 5.951e+00 2.235e+00
## 12 1.000 1.533e+01 2.116e+00
## 13 0.515 2.068e-01 3.283e-01
## 14 0.441 1.792e-01 2.944e-01
## 15 0.747 4.683e-01 4.984e-01
## 16 1.000 9.129e+03 2.014e+03
```

## Traffic Laws

- Speed Limit There are six dummy variables starting with sl, which indicate the speed limit mandated by the state for the year. The first four variables code speed limits in 5 mph increments from 55 to 75 mph. The fifth variable slnone indicates there was no speed limit for the state that year. The sixth variable sl70plus indicates that either the speed limit was 70 mph or greater, or that there was no speed limit that year.
- Seatbelts The next variable seatbelt is categorical and describes the type of seat belt law that exists, "0" if no law, '1' if primary (no other violation required to give a ticket), "2" if secondary (another violation must have occurred for the officer to issue a seatbelt ticket). There also exist two dummy variables starting with sb, one for primary and the other for secondary.
- Drinking The variables minage, zerotol, and bac describe the state's approach to drinking laws. The minage variable is the state's legal drinking age for the year, taking on 12 distinct values from ranging from 18 to 21, with 21 making up the great majority of observations. Non-integer observations indicate a year of The zerotol variable indicates if the state enacted a Zero Tolerance law for drinking, which makes it a criminal DUI offense for drivers under the age of 21 to drive with even a small amount of alcohol in their system. This variable takes on 11 distinct values, ranging from 0 to 1, with zero and 1 making up the vast majority of observations The next two bac dummy variables indicate if the state's acceptable BAC is 0.08% or 0.10%. Several states have adopted perse laws which allows for suspension or revocation of driver's license for DUI or DWI cases. The perse dummy variable indicates to the proportion of the year the state had this type of law enacted.
- Fatality Statistics Fatality statistics are given for each State-by-Year observation and include gross totals as well as totals normalized by vehicle miles driven by the state and per capita. These statistics are reported in terms of the time of occurrence, which include Fatalities over all times, Fatalities at nighttime, and Fatalities during the weekend.
- Demographics Observations also report a number of demographics specific to each state for each year reported. These include the number of vehicle miles in billions by the state's population for the year (total and per capita), the state's percent unemployment, and the percentage of the state's population between 14 and 24, inclusive.

As a validity check, we want to ensure that each year flag (identified and kept in d.vars) has the same number of observations (48 states). We can see that three states are missing - presumably the non-continental states (AK, DC, HI).

```
(data.desc[d.vars]) %>%
    sapply(function(v) v$counts) %>%
    t %>%
    data.frame %>%
    mutate_if(is.factor, as.character) %>%
    sapply(unique)
```

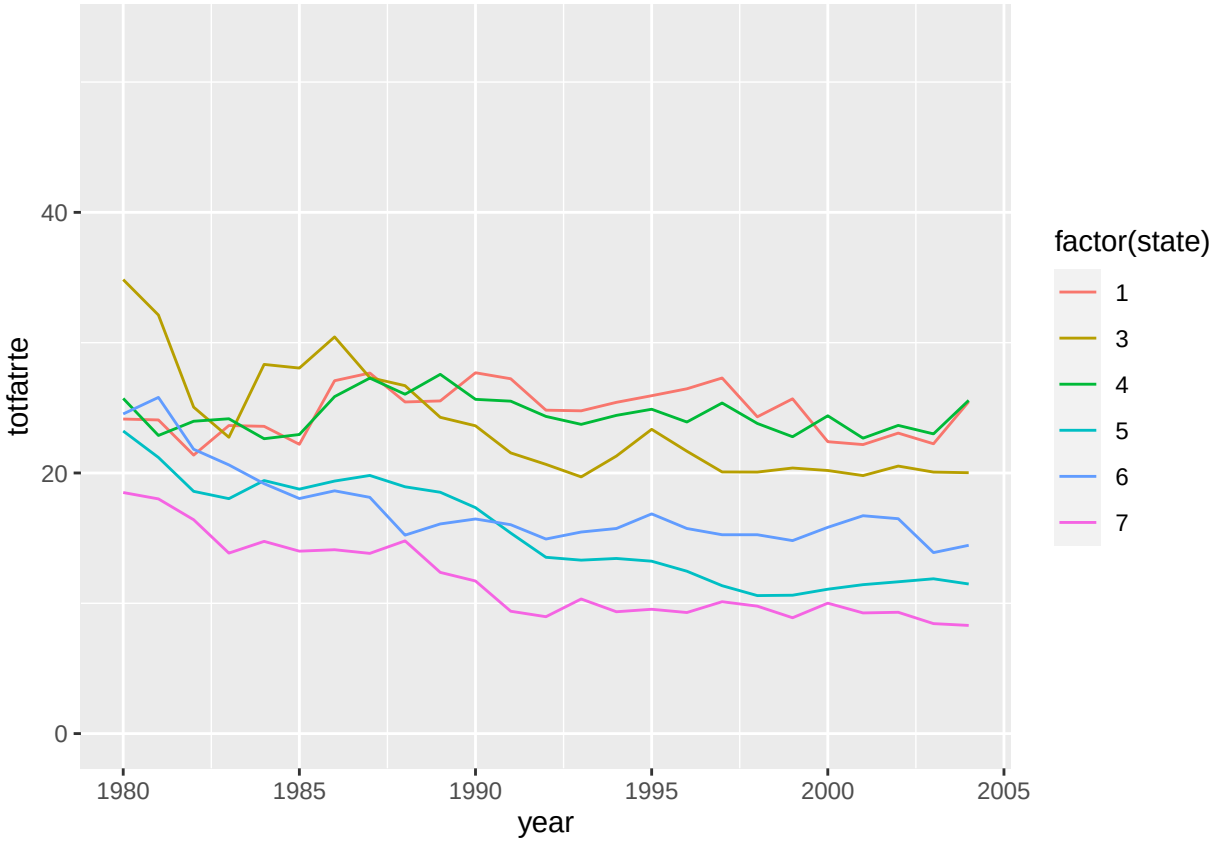
```
##         n   missing  distinct      Info       Sum      Mean       Gmd
##    "1200"       "0"       "2"   "0.115"      "48"    "0.04" "0.07686"
```

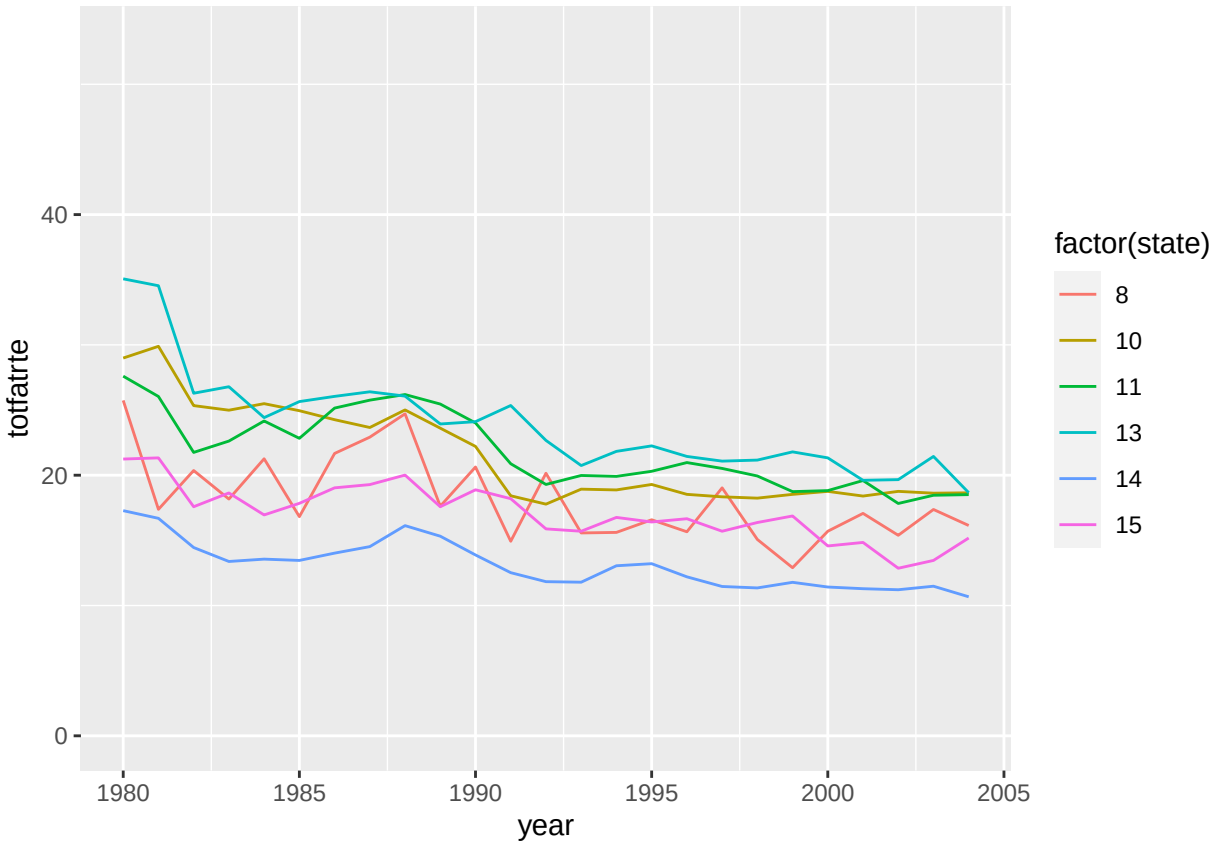
```
setdiff(1:51, unique(data$state))
```

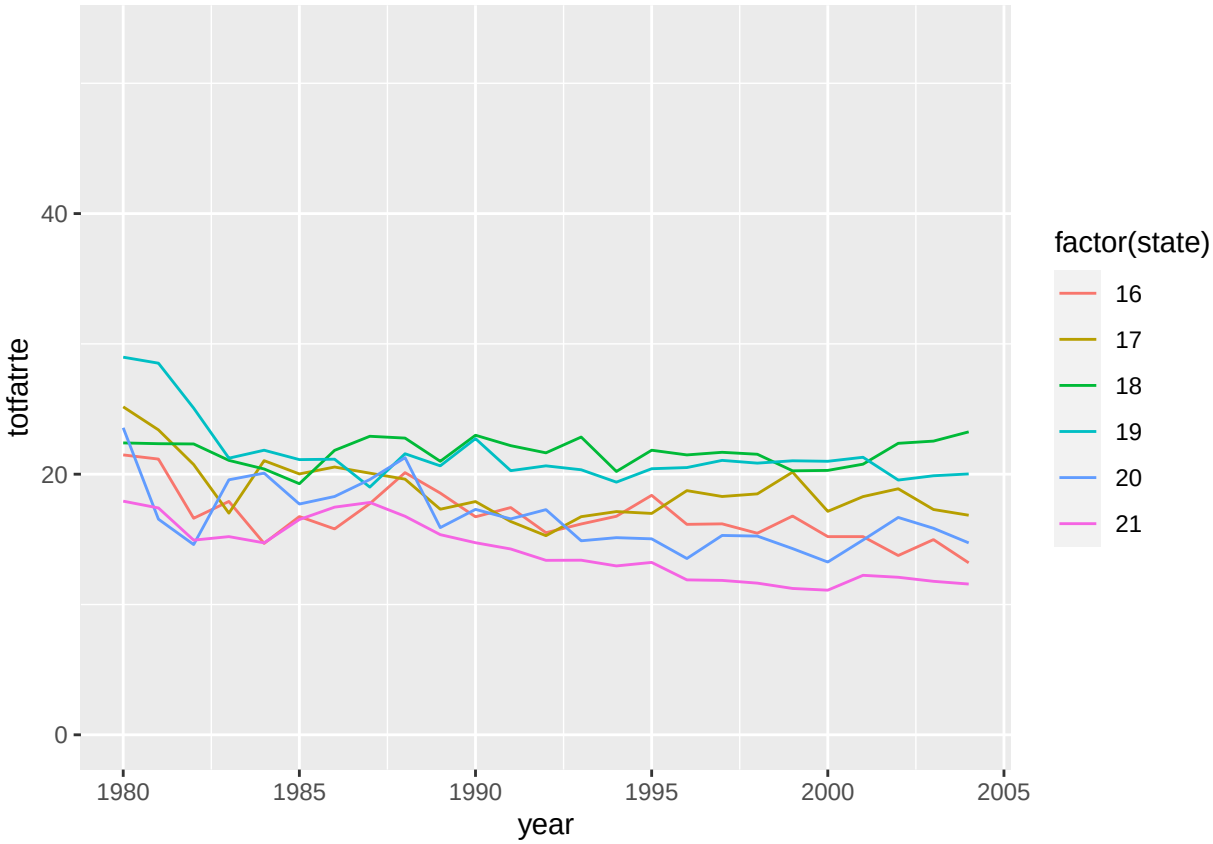
```
## [1]  2  9 12
```

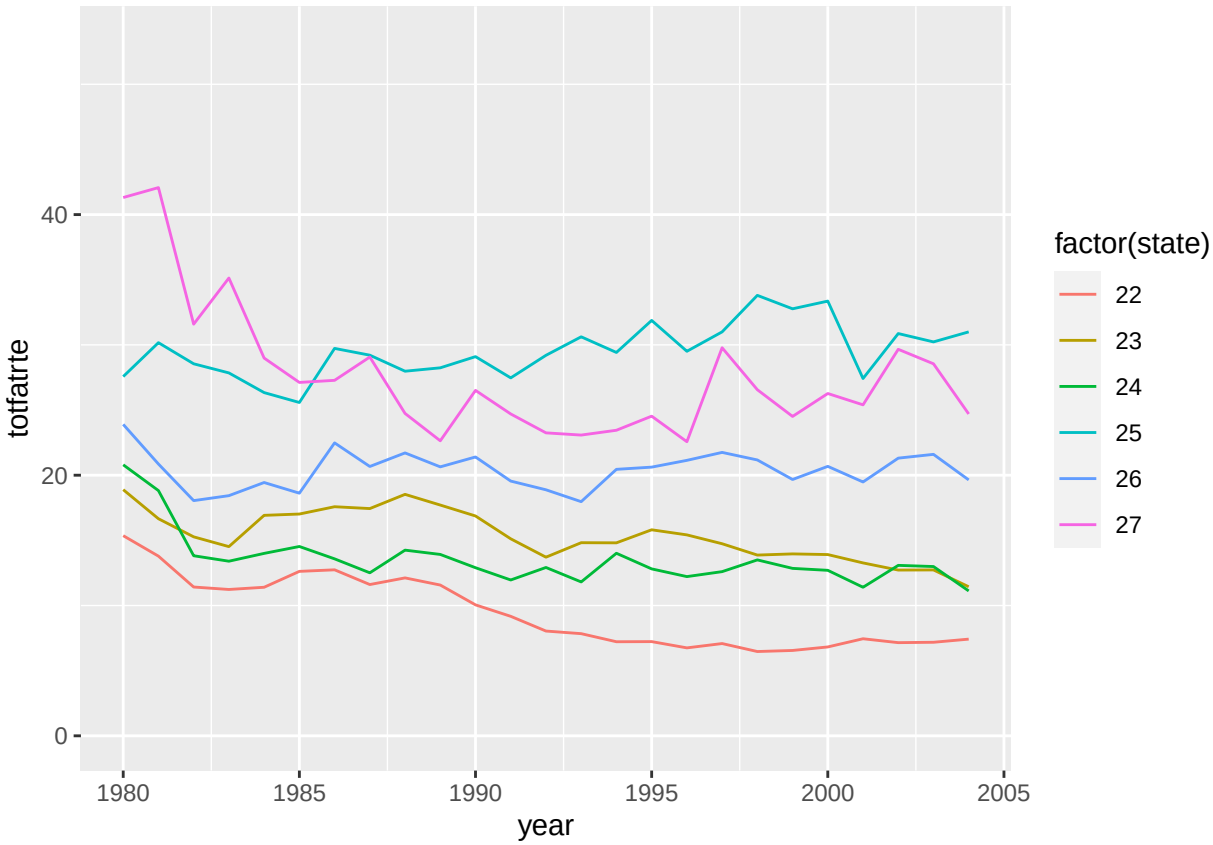
Next, we visualize trends in total fatality rate by state. We use a composite figure to allow observation of individual state trends without clutter. We can see that most states have a downwards trend in total fatality rate year-over-year.

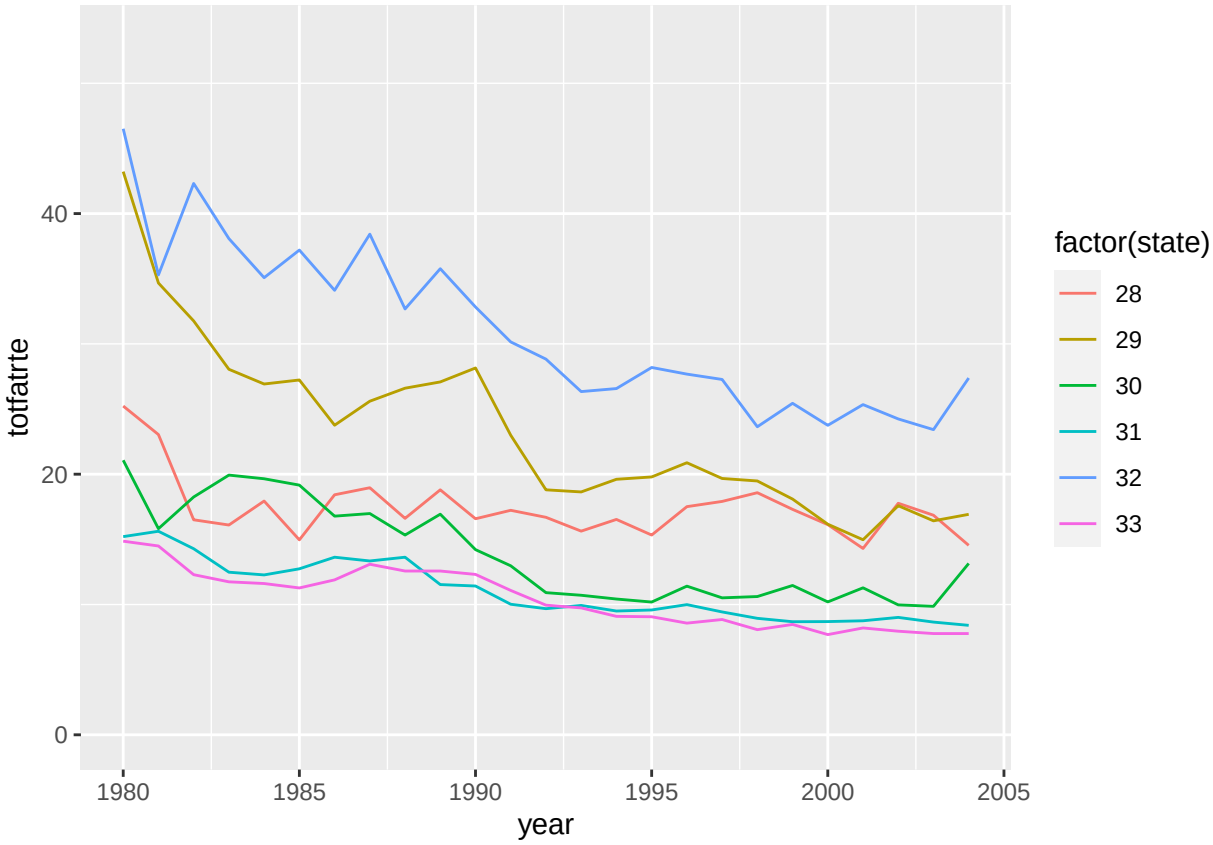
```
# cast(data[, c(m.vars, r.vars)], year~state)
ymax <- max(data$totfatrte)
for (state.batch in (unique(data$state) %>%
    split(., ceiling(seq_along(.)/6)))) {
    print(ggplot(subset(data, state %in% state.batch), aes(year, totfatrte)) + geom_line(aes(c
        ylim(0, ymax)) + theme(legend.position = "bottom")
}
```

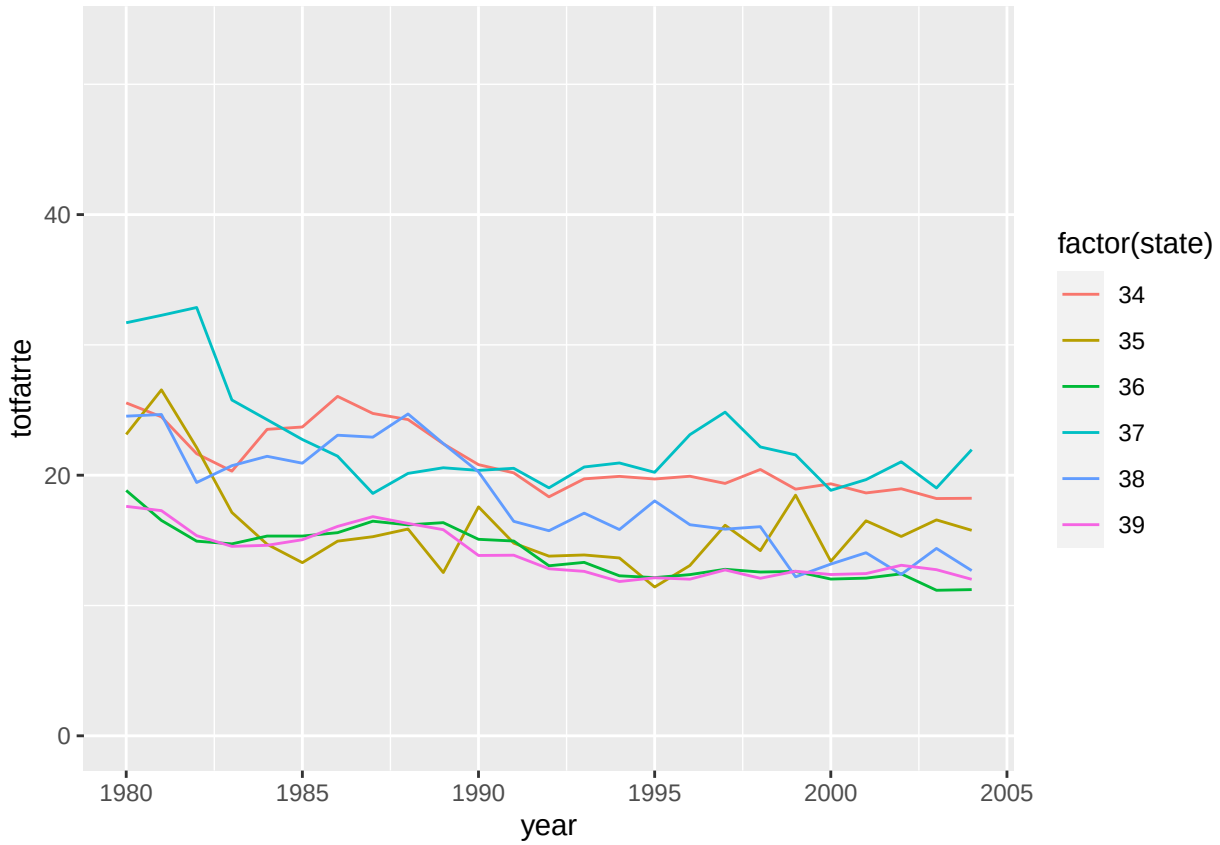


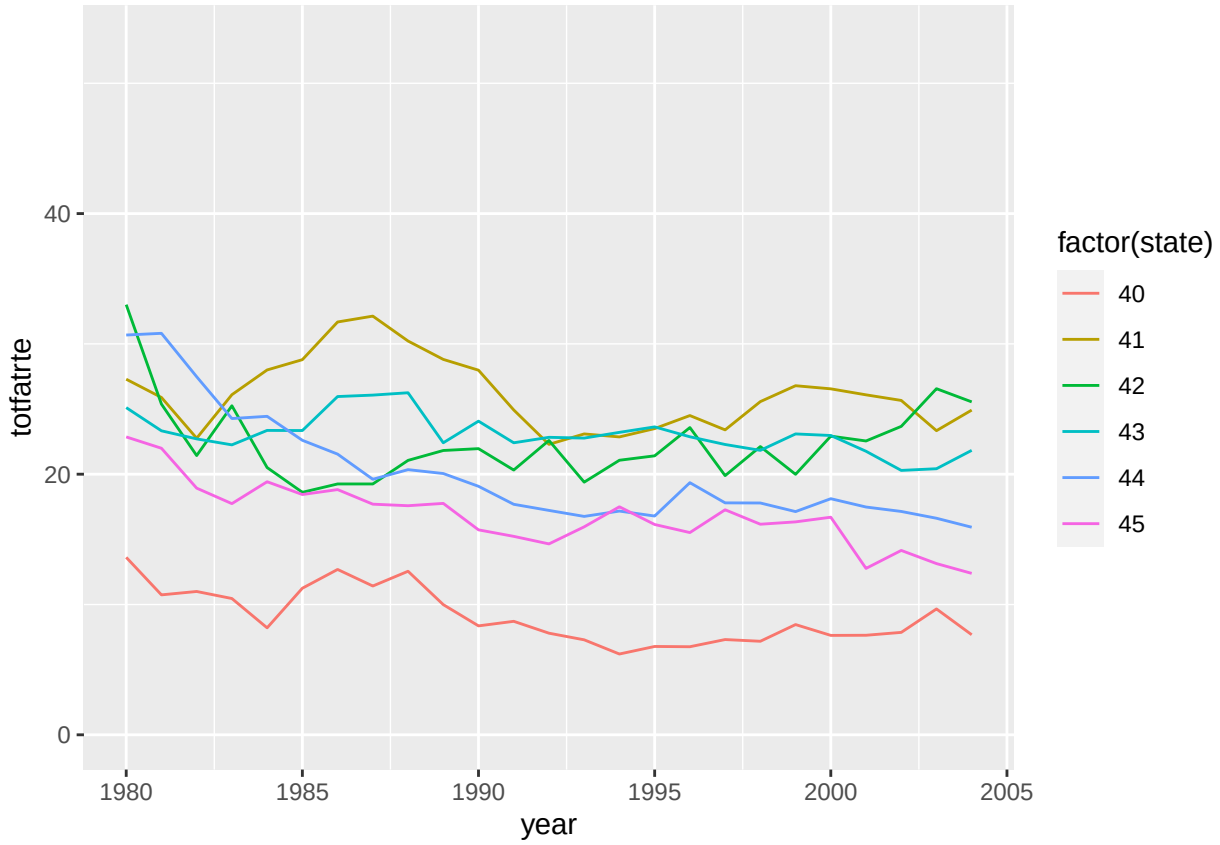

totfatrte
40
20
0
1980
1985
1990
1995
2000
2005
year
factor(state)
8
10
11
13
14
15

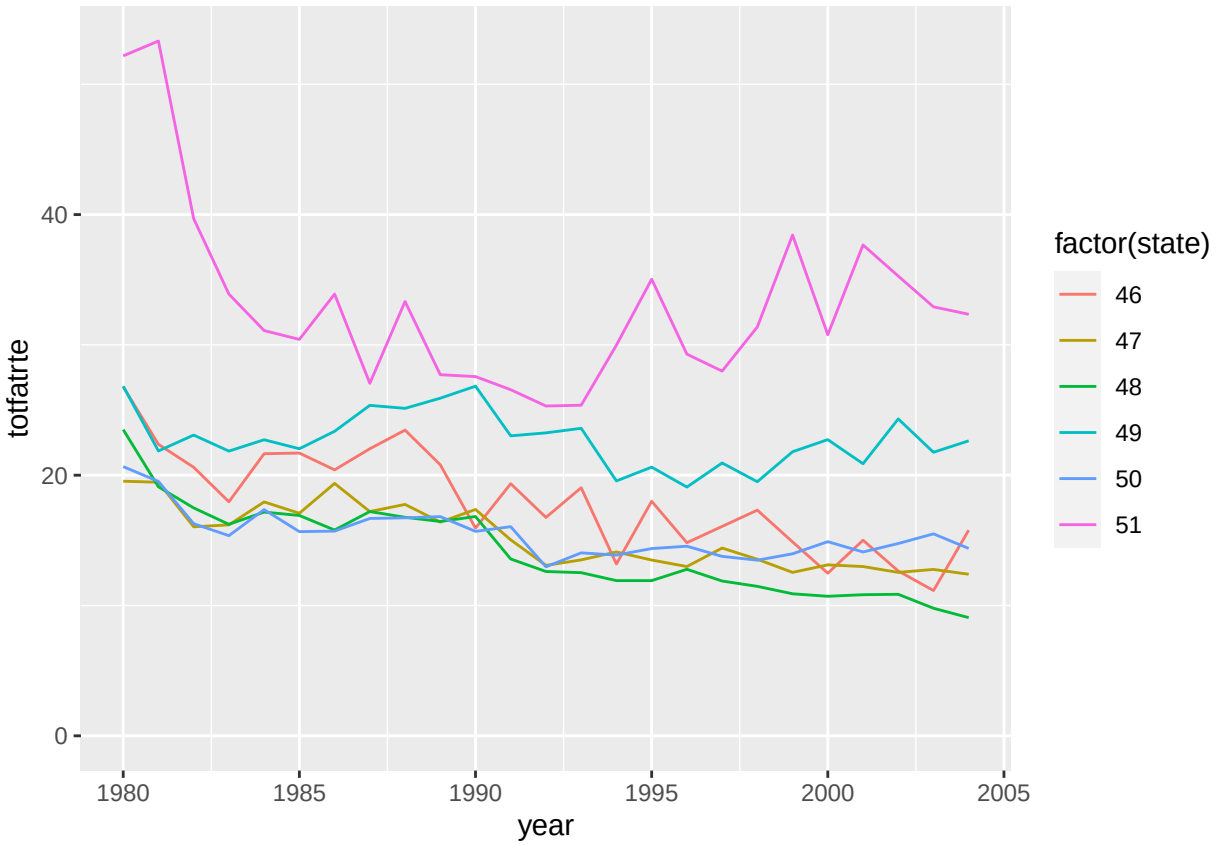

totfatrte
40
20
0
1980
1985
1990
1995
2000
2005
year
factor(state)
16
17
18
19
20
21


totfatrte
40
20
0
1980
1985
1990
1995
2000
2005
year
factor(state)
22
23
24
25
26
27


totfatrte
40
20
0
1980
1985
1990
1995
2000
2005
year
factor(state)
28
29
30
31
32
33


totfatrte
40
20
0
1980
1985
1990
1995
2000
2005
year
factor(state)
34
35
36
37
38
39


totfatrte
40
20
0
1980
1985
1990
1995
2000
2005
year
factor(state)
40
41
42
43
44
45




There are relatively few indicator values that are neither 0 nor 1, but these few could be problematic. If we treat the indicator as a factor, these values will introduce a number of distinct levels with small samples. If we treat this indicator as linear, there is an implication of linearity between the fractional year and the effect on total fatality rate. To reduce this effect, we will bucket all transition values (neither zero nor one) into a single representative value and then treat these variables as factors. Thus, 0 will indicate a complete lack of the indicator, 2 will indicate that the indicator was present throughout the entire year, and 1 will indicate a transition year where the indicator was true for part of the year. The functions defined below allow for conversions to this specified ternary factor as well as simple logical factors and a monthly factor.

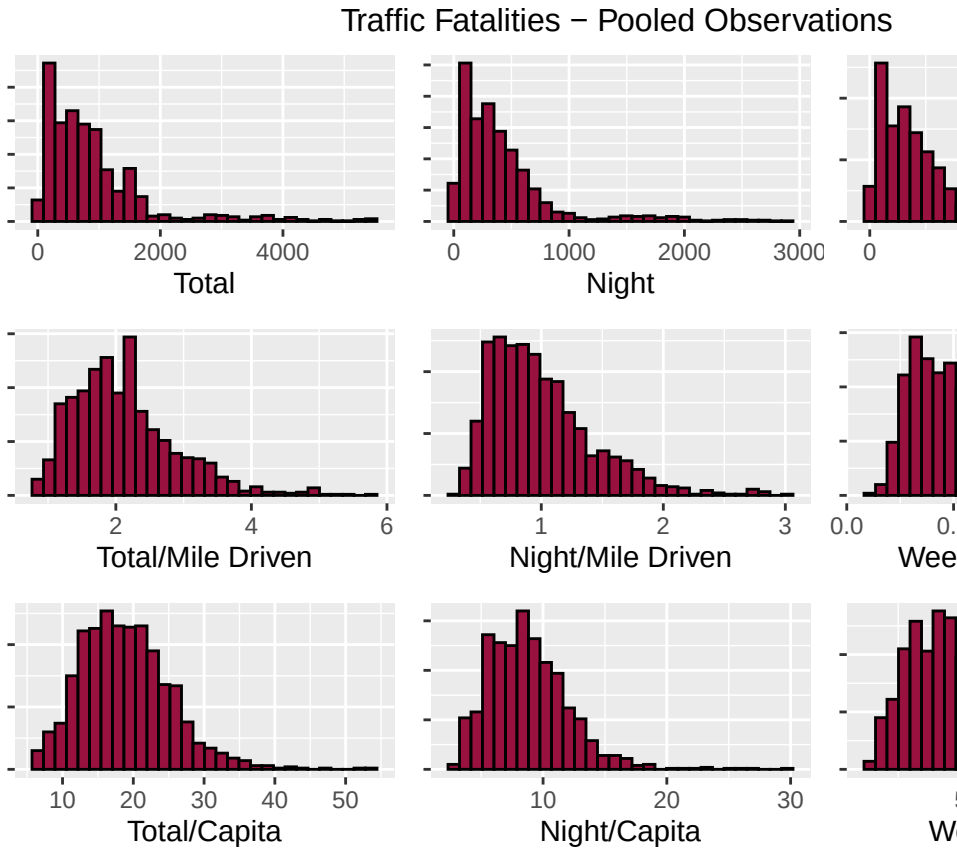
```
as.month.factor <- function(y) factor(round(12 * y), 0:12)
as.logical.factor <- function(b) factor(as.logical(b), c(TRUE, FALSE))
as.ternary.factor <- function(v) factor((v > 0) + (v == 1), c(0, 1, 2))
```

### Univariate EDA

```
traffic = data
h = geom_histogram(aes(y = ..count..), bins = 30, fill = "#99123F", colour = "black")
t = theme(plot.title = element_text(lineheight = 1, face = "bold"), axis.text.y = element_blank
    axis.title.y = element_blank())

plot.hist1 = ggplot(traffic, aes(x = totfat)) + scale_x_continuous(name = "Total") + h + t
plot.hist2 = ggplot(traffic, aes(x = nghtfat)) + scale_x_continuous(name = "Night") + h + t
plot.hist3 = ggplot(traffic, aes(x = wkndfat)) + scale_x_continuous(name = "Weekend") + h +
```

```
    t + t
plot.hist4 = ggplot(traffic, aes(x = totfatpvm)) + scale_x_continuous(name = "Total/Mile Driver
    h + t
plot.hist5 = ggplot(traffic, aes(x = nghtfatpvm)) + scale_x_continuous(name = "Night/Mile Drive
    h + t + t
plot.hist6 = ggplot(traffic, aes(x = wkndfatpvm)) + scale_x_continuous(name = "Weekend/Mile Dri
    h + t + t
plot.hist7 = ggplot(traffic, aes(x = totfatrte)) + scale_x_continuous(name = "Total/Capita") +
    h + t + t
plot.hist8 = ggplot(traffic, aes(x = nghtfatrte)) + scale_x_continuous(name = "Night/Capita") +
    h + t + t
plot.hist9 = ggplot(traffic, aes(x = wkndfatrte)) + scale_x_continuous(name = "Weekend/Capita")
    h + t
grid.arrange(plot.hist1, plot.hist2, plot.hist3, plot.hist4, plot.hist5, plot.hist6, plot.hist7
    plot.hist8, plot.hist9, nrow = 3, ncol = 3, top = quote("Traffic Fatalities - Pooled Observ
```




**Dependent Variables - Fatalities**

```
shapiro.test(traffic$totfat)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  traffic$totfat
## W = 0.75, p-value <2e-16
```

```
shapiro.test(traffic$totfatpvm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  traffic$totfatpvm
## W = 0.94, p-value <2e-16
```

```
shapiro.test(traffic$totfatrte)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  traffic$totfatrte
## W = 0.97, p-value = 2e-15
```

```
shapiro.test(log(traffic$totfat))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(traffic$totfat)
## W = 0.98, p-value = 1e-11
```

```
shapiro.test(log(traffic$totfatpvm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(traffic$totfatpvm)
## W = 1, p-value = 0.04
```

```
shapiro.test(log(traffic$totfatrte))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(traffic$totfatrte)
## W = 0.99, p-value = 9e-07
```

We see that pooling the State-Year observations underscores meaningful information about the underlying trends of the fatality variables. Skew is most evident for the unnormalized fatality rates (Total, Night, and Weekend) and least evident when the data is normalized by the state's population. We see that virtually every variable is positively skewed, which means a log transformation may improve normality. When considering normalization techniques (none, per vehicle mile, per capita), it appears that $totfatpvm$ responds most favorably to the log transformation.

##Univariate Analysis of DVs by Year

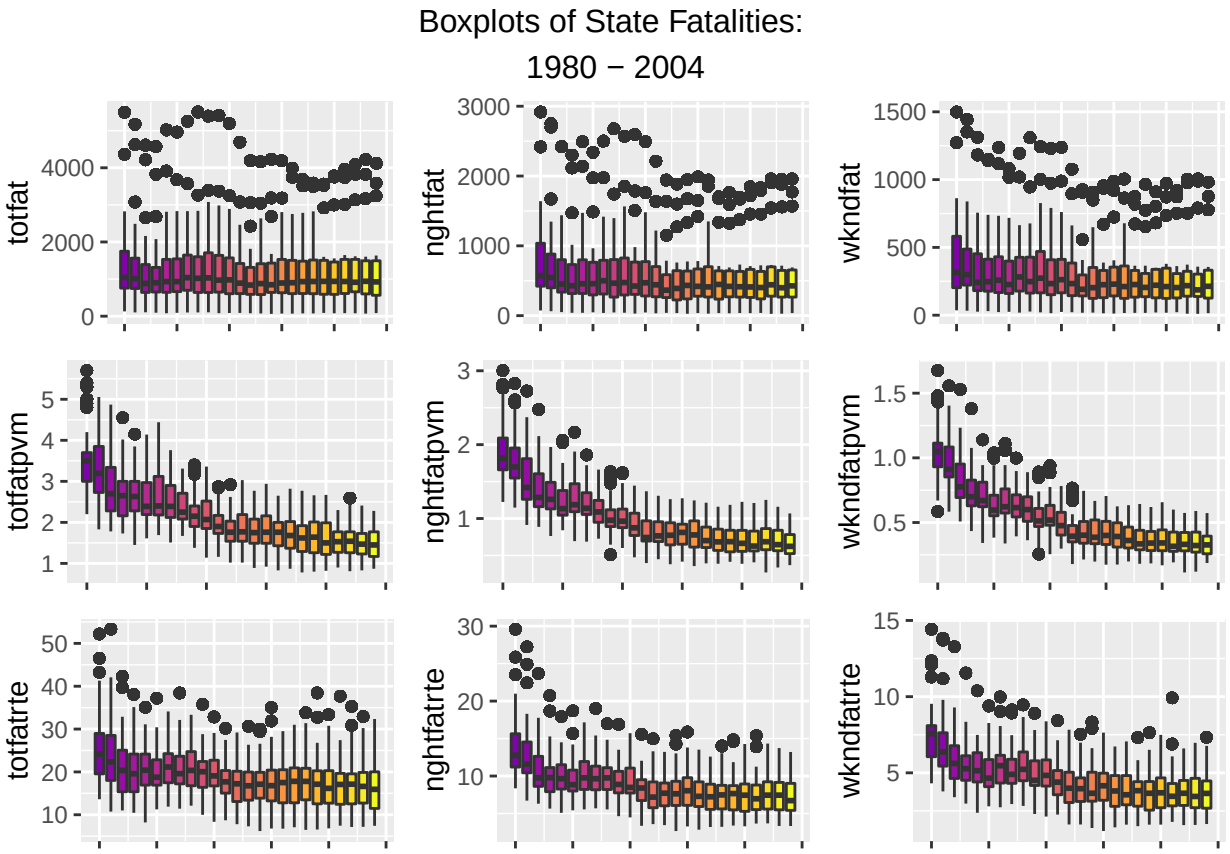
```
# Shape files for each state
states = map_data("state", projection = "albers", parameters = c(39, 45))
```

```r
# Associate State Index with State Name
statenames = unique(states$region[])
statenames = c(statenames[1], "alaska", statenames[2:10], "hawaii", statenames[11:length(state
states = states[states$region != "district of columbia", ]
states$state = match(states$region, statenames)
# Associate the state shapes with associated observations
traffic.map = merge(states, traffic, by = "state")


b = geom_boxplot(aes(fill = year, group = year))
t = theme(plot.title = element_text(lineheight = 1, face = "bold"), legend.position = "none",
    axis.text.x = element_blank(), axis.title.x = element_blank())

library(viridis)
c = scale_fill_viridis(option = "plasma", begin = 0.25)

plot.bp1 = ggplot(traffic.map, aes(year, totfat)) + b + t + c
plot.bp2 = ggplot(traffic.map, aes(year, nghtfat)) + b + t + c
plot.bp3 = ggplot(traffic.map, aes(year, wkndfat)) + b + t + c
plot.bp4 = ggplot(traffic.map, aes(year, totfatpvm)) + b + t + c
plot.bp5 = ggplot(traffic.map, aes(year, nghtfatpvm)) + b + t + c
plot.bp6 = ggplot(traffic.map, aes(year, wkndfatpvm)) + b + t + c
plot.bp7 = ggplot(traffic.map, aes(year, totfatrte)) + b + t + c
plot.bp8 = ggplot(traffic.map, aes(year, nghtfatrte)) + b + t + c
plot.bp9 = ggplot(traffic.map, aes(year, wkndfatrte)) + b + t + c
grid.arrange(plot.bp1, plot.bp2, plot.bp3, plot.bp4, plot.bp5, plot.bp6, plot.bp7, plot.bp8,
    plot.bp9, nrow = 3, ncol = 3, top = quote("Boxplots of State Fatalities: \n1980 - 2004"))
```

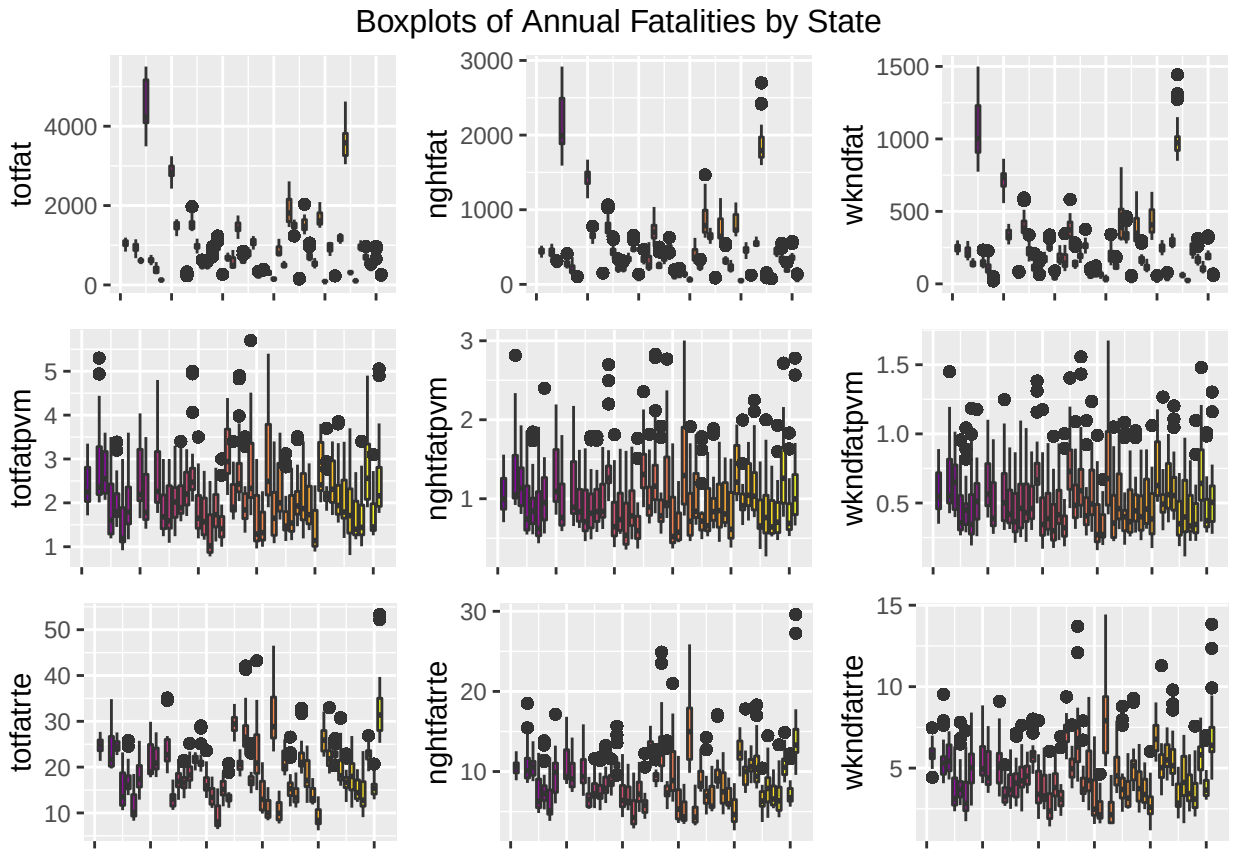



**Total:** Top outliers look to have decreased over time, but the national average of gross fatalities looks almost unchanged over time. **PVM:** Significant decline over time in average rate per vehicle mile driven. Variance looks to be proportional to mean, also decreasing over time. **RTE:** Significant decline over time in average fatalities per capita. Variance does not look to be proportional to mean.

##Univariate Analysis of DVs by State

```
b = geom_boxplot(aes(fill = state, group = state))
t = theme(plot.title = element_text(lineheight = 1, face = "bold"), legend.position = "none",
    axis.text.x = element_blank(), axis.title.x = element_blank())
library(viridis)
c = scale_fill_viridis(option = "plasma", begin = 0.25)

plot.bp1 = ggplot(traffic.map, aes(state, totfat)) + b + t + c
plot.bp2 = ggplot(traffic.map, aes(state, nghtfat)) + b + t + c
plot.bp3 = ggplot(traffic.map, aes(state, wkndfat)) + b + t + c
plot.bp4 = ggplot(traffic.map, aes(state, totfatpvm)) + b + t + c
plot.bp5 = ggplot(traffic.map, aes(state, nghtfatpvm)) + b + t + c
plot.bp6 = ggplot(traffic.map, aes(state, wkndfatpvm)) + b + t + c
plot.bp7 = ggplot(traffic.map, aes(state, totfatrte)) + b + t + c
plot.bp8 = ggplot(traffic.map, aes(state, nghtfatrte)) + b + t + c
plot.bp9 = ggplot(traffic.map, aes(state, wkndfatrte)) + b + t + c
grid.arrange(plot.bp1, plot.bp2, plot.bp3, plot.bp4, plot.bp5, plot.bp6, plot.bp7, plot.bp8,
    plot.bp9, nrow = 3, ncol = 3, top = "Boxplots of Annual Fatalities by State")
```




There is good agreement between state and rates when considering timing of the incident (overall, at night, and during the weekend).

**Total Fatalities:** Variance is proportional to the mean of the state, which implies using these variables could result in significant heteroskedasticity. There are also a few outliers that would significantly affect the results (likely California and Texas). **Per Vehicle Mile:** Observations look constant with similar variance and no major outliers that could influence linear modeling.
**Per Capita:** No states that seem to be outliers, variance is not constant between states, nor a function of average.

##Univariate Choropleths of DVs

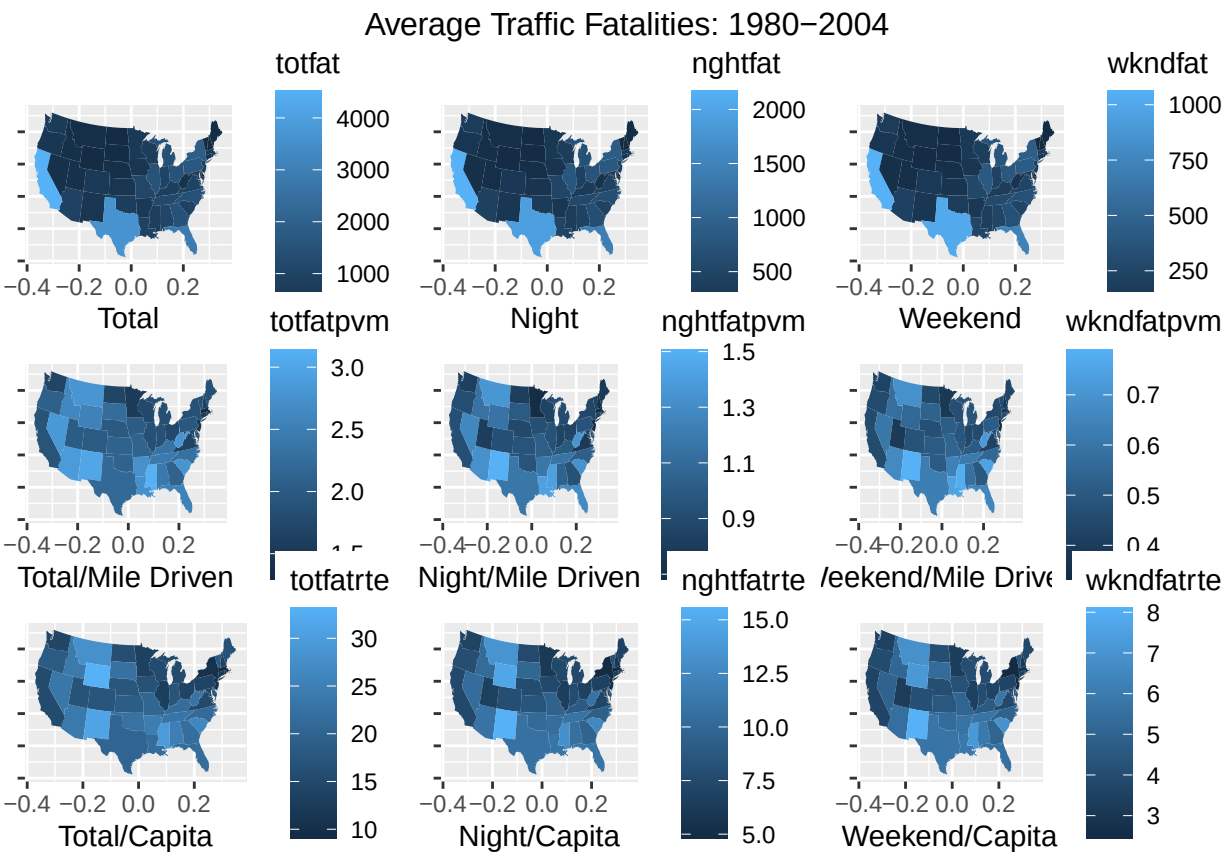
```
states = map_data("state", projection = "albers", parameters = c(39, 45))
statenames = unique(states$region[])
statenames = c(statenames[1], "alaska", statenames[2:10], "hawaii", statenames[11:length(state
states = states[states$region != "district of columbia", ]
states$state = match(states$region, statenames)
traffic.map = merge(states, traffic, by = "state")

no_var = !names(traffic.map) %in% c("year", "region", "subregion", "lat", "long", "order",
    "group")
traffic.state.agg = aggregate(traffic.map[, no_var], list(traffic.map$state), mean)
traffic.map.agg = merge(states, traffic.state.agg, by = "state")

t = theme(plot.title = element_text(lineheight = 1, face = "bold"), axis.text.y = element_blan
```

```
    axis.title.y = element_blank())

plot.map1 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = totfat, group = g
    labs(x = "Total") + t
plot.map2 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = nghtfat, group =
    labs(x = "Night") + t
plot.map3 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = wkndfat, group =
    labs(x = "Weekend") + t
plot.map4 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = totfatpvm, group
    labs(x = "Total/Mile Driven") + t
plot.map5 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = nghtfatpvm, grou
    labs(x = "Night/Mile Driven") + t
plot.map6 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = wkndfatpvm, grou
    labs(x = "Weekend/Mile Driven") + t
plot.map7 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = totfatrte, group
    labs(x = "Total/Capita") + t
plot.map8 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = nghtfatrte, grou
    labs(x = "Night/Capita") + t
plot.map9 = qplot(long, lat, data = traffic.map.agg, geom = "polygon", fill = wkndfatrte, grou
    labs(x = "Weekend/Capita") + t
grid.arrange(plot.map1, plot.map2, plot.map3, plot.map4, plot.map5, plot.map6, plot.map7, plot
    plot.map9, nrow = 3, ncol = 3, top = quote("Average Traffic Fatalities: 1980-2004\n"))
```




**TOTAL:** California, Texas, and Florida have the highest overall total fatalities averaged over

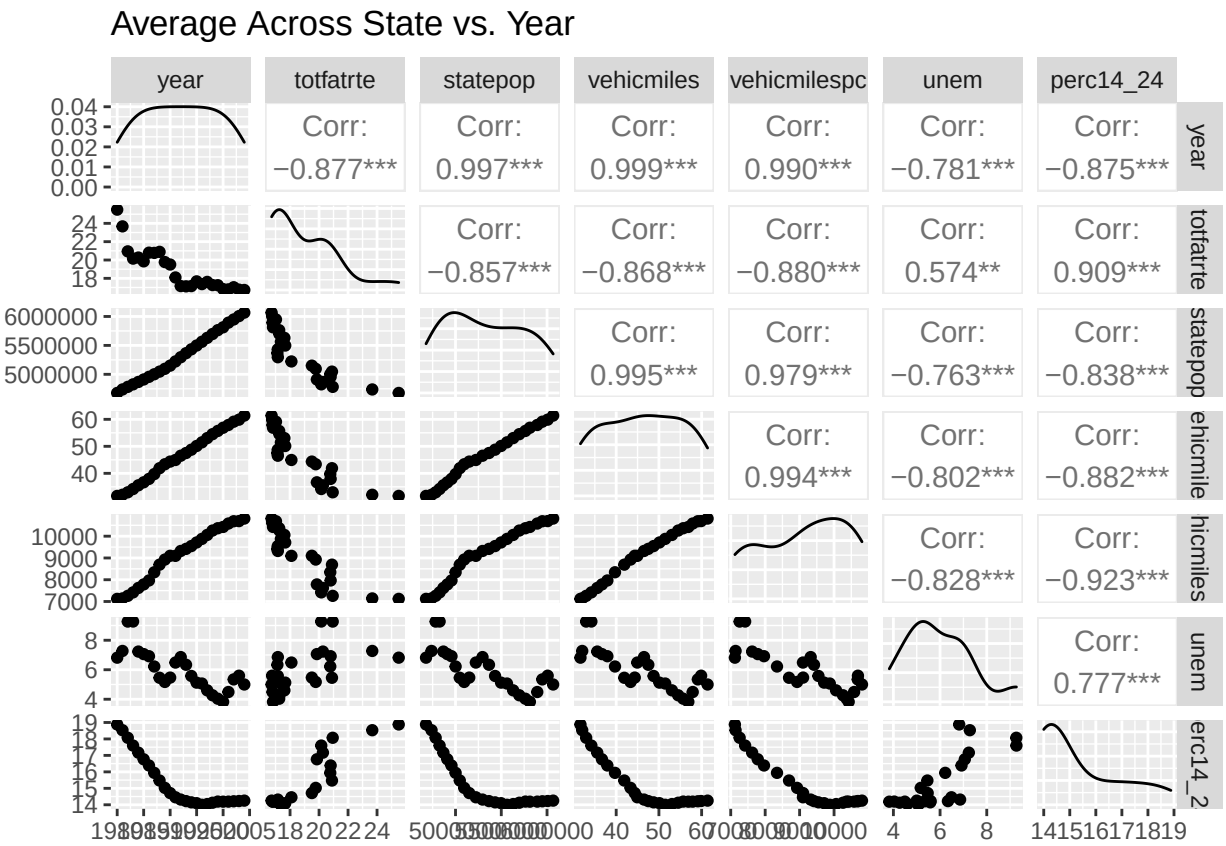
all years observed. It's apparent that these observations are outliers and will skew our modeling. **PVM:** Evidence of regional influence. Significantly higher rates just inland from West Coast and in Lousisiana/Mississippi. Perhaps State GDP per Capita might be an unexplained factor? **RTE:** Wyoming and Montana show higher per capita rates. There is some regional influence but probably less spatial correlation as compared to pvm.

**Overall Univariate Conclusions:** The *totfatrte* variable is not a bad choice to use as our dependent variable. Of the variables available, it looks most normally distributed when pooled, doesn't have many influential outliers, nor does there seem to be regional correlation, which could contribute to omitted variable bias. There also doesn't look to be much serial correlation or heteroskedasticity over time, which means we may be able to get reasonable results from a pooled OLS.

### Bivariate EDA

####Average of State Observations Over Time

```
mean_by_year = aggregate(traffic[, c("totfatrte", "statepop", "vehicmiles", "vehicmilespc",
    "unem", "perc14_24")], traffic["year"], FUN = mean)
ggpairs(mean_by_year) + ggtitle(label = "Average Across State vs. Year")
```




We see significant time dependence between each variable and time when we average all state observations. The fatality rate goes down, population increases and gross vehicle miles driven increase. Unemployment is cyclical, but shows a steady decline overall.

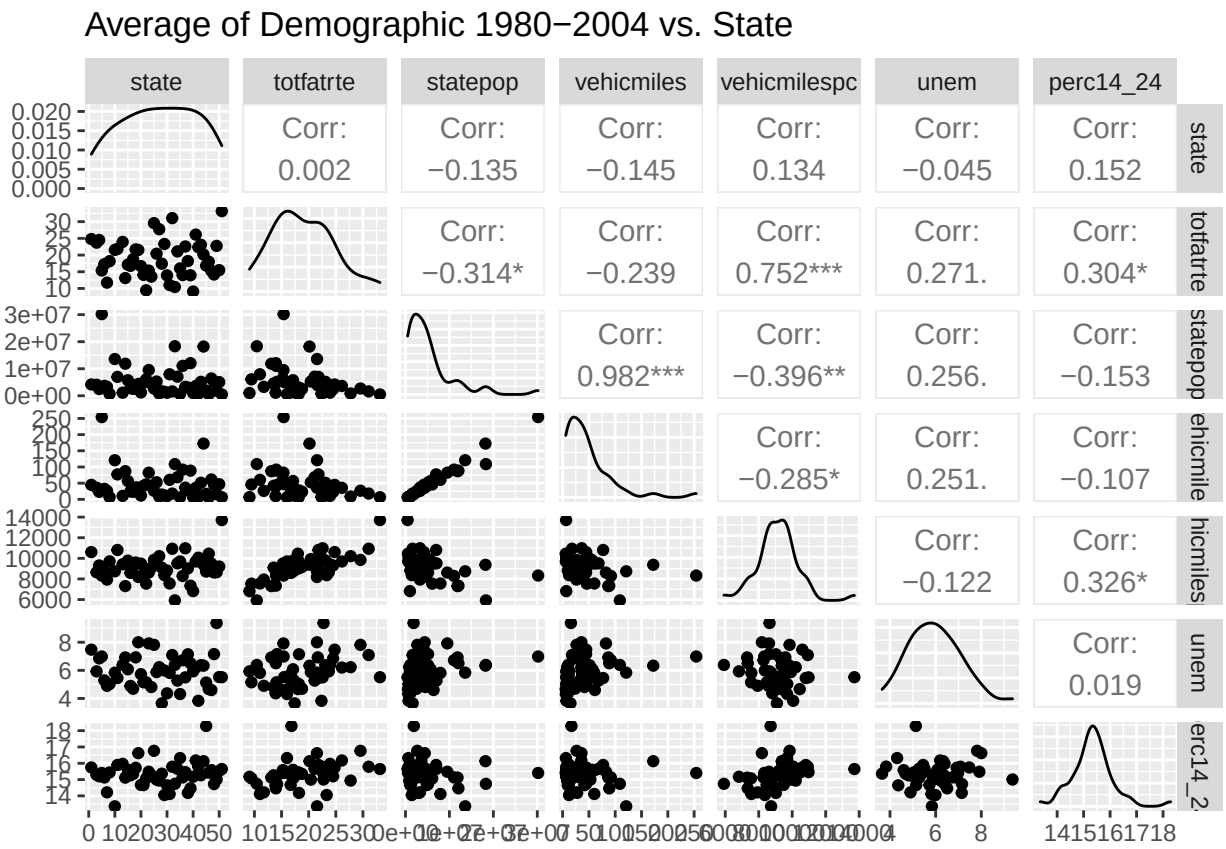
Miles driven per capita increases, which is not necessarily expected. This could be attributed to the fact that cars have become more affordable over time and are therefore more accessible. Percent

14_24 decreases, which indicate an aging population and might mean there are more experienced drivers and therefore fewer accidents. For this reason, this might be a good variable to include in the final model. Beyond this, when comparing scatters of any two variables, we cannot consider their covariance as significant due to spurious correlation.

####Year-Aggregates By State

```
mean_by_state = aggregate(traffic[, c("totfatrte", "statepop", "vehicmiles", "vehicmilespc",
    "unem", "perc14_24")], traffic["state"], FUN = mean)
ggpairs(mean_by_state, size = 12) + ggtitle(label = "Average of Demographic 1980-2004 vs. State
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'size' are being
## ignored. If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```



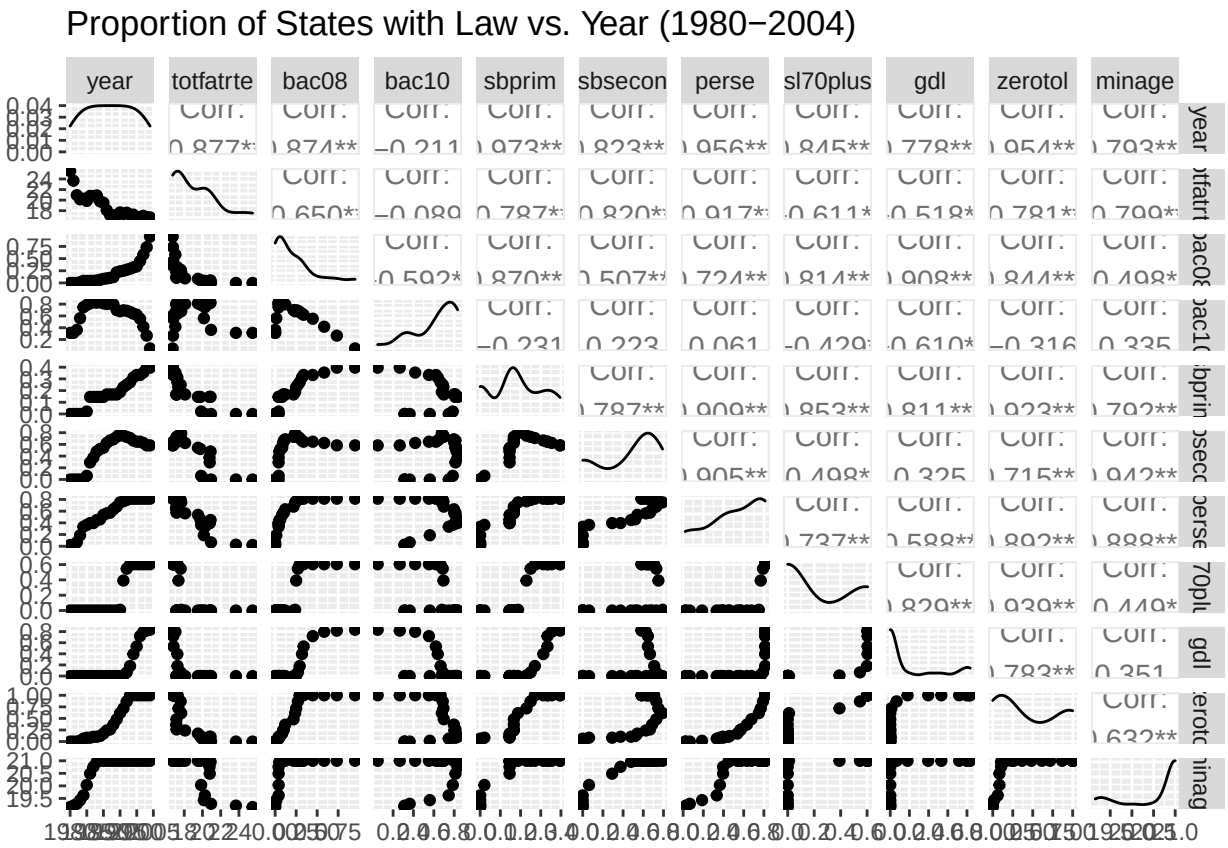

When considering average of demographic variables from 1980-2004 vs. State, we see the most significant correlation between vehicle miles driven and population. This is not surprising, since more people to drive means more miles driven. A significant observation is the correlation between vehicle miles per capita and fatality rate. This makes sense because more time on the road means greater chance of accident. Negative correlation between state population and totfatrte. This is unexpected, but might be explained by more people in a state, which means more tax revenue, more investment in safe driving infrastructure. We see that vehicle miles pc go down with state population as well, meaning that more populous states tend to travel on the road less, thereby putting themselves at lower risk of fatality. We see a medium correlation with youth and totfatrte. Certainly, the higher the proportion of youth, the more likely we are to have youth accidents, but there is also a medium effect for youth and miles pc. Younger states drive more, which could

explain higher rate of accidents, not necessarily that younger drivers are worse at driving (although they probably are).

## Fatality Rate and Traffic Laws: State-Aggregates Over Time:

```
mean_by_year = aggregate(traffic[, c("totfatrte", "bac08", "bac10", "sbprim", "sbsecon", "perse
    "sl70plus", "gdl", "zerotol", "minage")], traffic["year"], FUN = mean)
ggpairs(mean_by_year, size = 3) + ggtitle(label = "Proportion of States with Law vs. Year (198
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'size' are being
## ignored. If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```

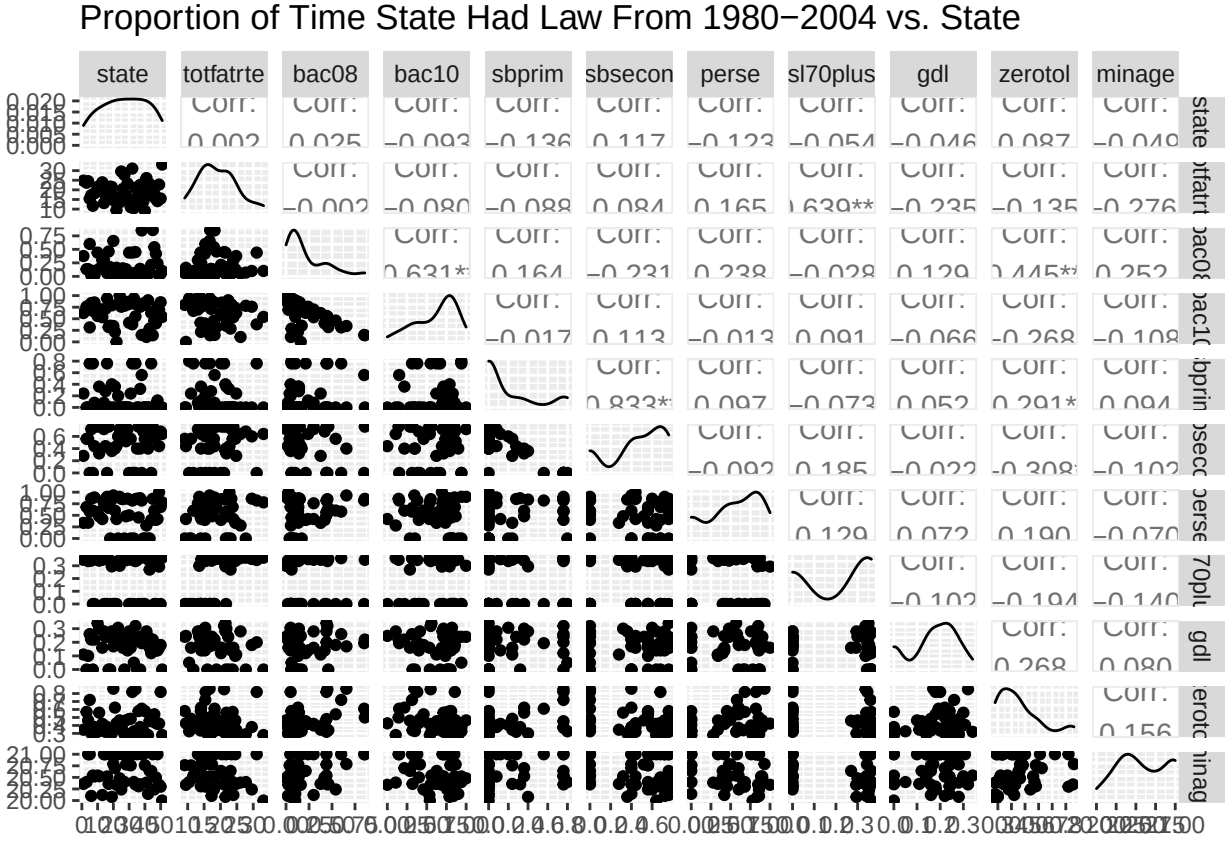



These data points indicate trends for the entire country and give us a sense of the adoption of each law over time. We see a strong time dependence between each of the law variables and time. Although the correlation between time and *bac*10 is low, it's evident that a quadratic term would do an excellent job of predicting the number of states with the law at any given year. When comparing scatters of any two variables, we cannot consider their covariance as significant due to spurious correlation. However, there There seems to be strong covariance between the *gdl* and *sbprim* covariates. That is to say, for a given year, if we see a relatively high number of states with *gdl* laws, we would also expect there to be a relatively high number of states with *sbprim* laws for that year.

**Fatality Rate and Traffic Laws: Year-Aggregates By State** #"'{r out.height="400px", out.width="400px",fig.height=10,fig.width=10}

```
mean_by_state = aggregate(traffic[, c("totfatrte", "bac08", "bac10", "sbprim", "sbsecon", "pers
    "sl70plus", "gdl", "zerotol", "minage")], traffic["state"], FUN = mean)
ggpairs(mean_by_state, size = 3) + ggtitle(label = "Proportion of Time State Had Law From 1980-
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: 'size' are being
## ignored. If these are meant to be aesthetics, submit them using the 'mapping'
## variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```



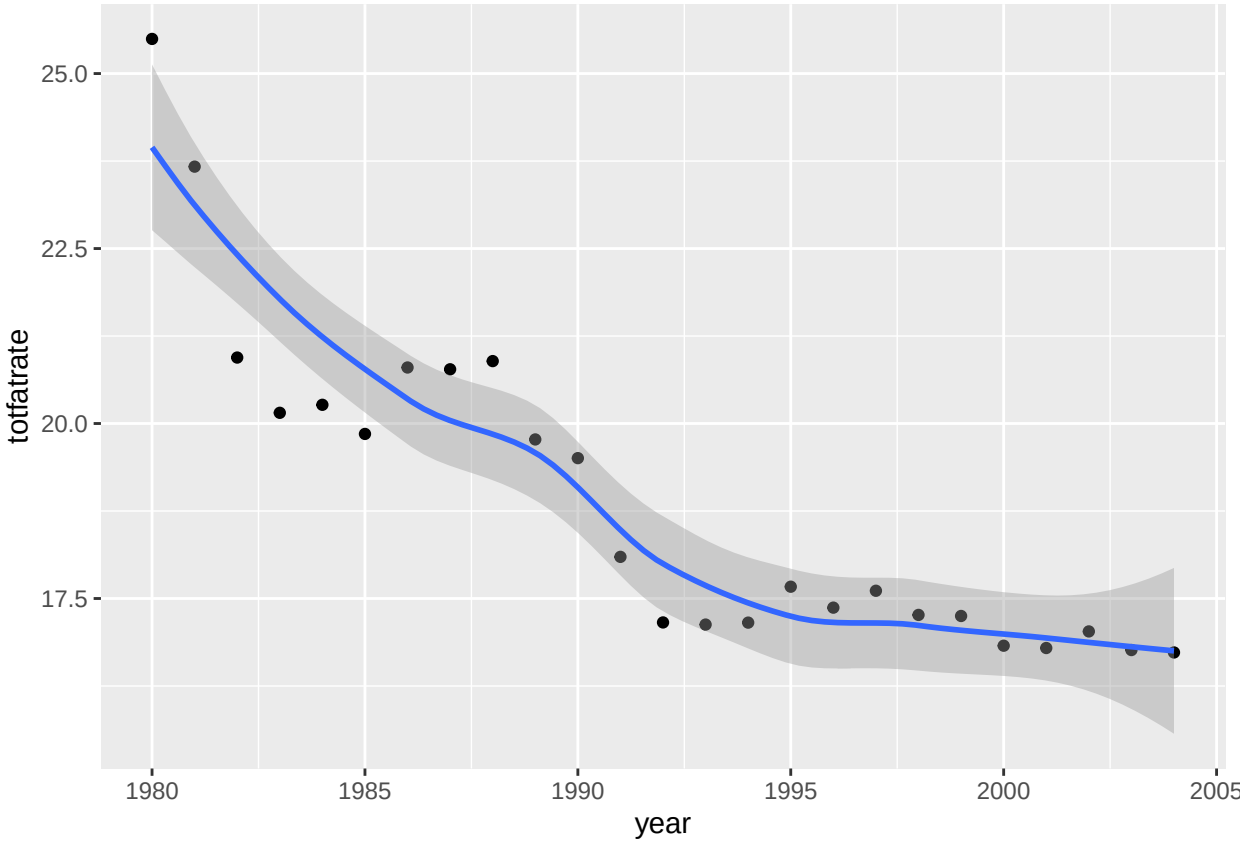

There is strong correlation between *bac*08 and *bac*10 variables and *sbprim* and *sbsecon* variables. This makes sense, because these sets of laws are mutually exclusive; a state can only have one law or the other. Therefore, we would expect to see that the more years a state has one variable, the less years we would expect to see of its converse law. There are a few cases of significant interaction between covariates. For example, the correlation between *bac*08 and *zerotol* is 0.445. This means that the longer a state has had a *bac*08 law passed, the longer we would expect that state to have had a *zerotol* law passed as well.

### Average Overall Fatality Rate:

We start by plotting the average fatality rate(over all 48 states) by year. We can see that over time we have observed a decline in the total fatalities per 100,000 population. Without controlling for any factors, driving has become safer over time.

```
library(plyr)
yearly_fatality <- ddply(data, .(year), summarise, totfatrate = mean(totfatrte))
ggplot(data = yearly_fatality, mapping = aes(x = year, y = totfatrate)) + geom_point() + geom_s
```

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




We now create an initial model of fatality rate on yearly indicator variables only.

```
(simple.formula <- d.vars %>%
    c(list(sep = "+")) %>%
    do.call(paste, .) %>%
    paste0(r.vars, "~", .) %>%
    as.formula)
```

```
## totfatrte ~ d80 + d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 +
##     d89 + d90 + d91 + d92 + d93 + d94 + d95 + d96 + d97 + d98 +
##     d99 + d00 + d01 + d02 + d03 + d04
## <environment: 0x7ff46f0584f8>
```

```
(simple.lm <- lm(simple.formula, data = data)) %>%
    summary
```

```
##
## Call:
## lm(formula = simple.formula, data = data)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.930  -4.347  -0.731   3.749  29.650
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  16.7290     0.8671   19.29  < 2e-16 ***
## d80           8.7656     1.2263    7.15  1.5e-12 ***
## d81           6.9412     1.2263    5.66  1.9e-08 ***
## d82           4.2135     1.2263    3.44  0.00061 ***
## d83           3.4240     1.2263    2.79  0.00532 **
## d84           3.5385     1.2263    2.89  0.00398 **
## d85           3.1225     1.2263    2.55  0.01101 *
## d86           4.0715     1.2263    3.32  0.00093 ***
## d87           4.0458     1.2263    3.30  0.00100 ***
## d88           4.1627     1.2263    3.39  0.00071 ***
## d89           3.0433     1.2263    2.48  0.01321 *
## d90           2.7763     1.2263    2.26  0.02376 *
## d91           1.3658     1.2263    1.11  0.26560
## d92           0.4290     1.2263    0.35  0.72655
## d93           0.3987     1.2263    0.33  0.74511
## d94           0.4263     1.2263    0.35  0.72821
## d95           0.9396     1.2263    0.77  0.44371
## d96           0.6404     1.2263    0.52  0.60160
## d97           0.8817     1.2263    0.72  0.47230
## d98           0.5365     1.2263    0.44  0.66186
## d99           0.5215     1.2263    0.43  0.67075
## d00           0.0967     1.2263    0.08  0.93718
## d01           0.0637     1.2263    0.05  0.95855
## d02           0.3006     1.2263    0.25  0.80638
## d03           0.0346     1.2263    0.03  0.97751
## d04               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.01 on 1175 degrees of freedom
## Multiple R-squared:  0.128,  Adjusted R-squared:  0.11
## F-statistic: 7.16 on 24 and 1175 DF,  p-value: <2e-16
```

This model explains what the effect of each year on total fatality rate is expected to be. The coefficients associated with each of these yearly indicators is declining as the year dummy increases - which is expected, since our plot of average fatality rate over the years shows a declining trend. We do note that the coefficients of earlier years all achieve statistical significance (up to 1990), while all yearly coefficients after 1990 are non-significant. This is due to the fact that the standard error for each of the estimates remains the same while the estimates decline in magnitude. The regression explains how the fatality rate has changed (average across the 48 states) as compared to the year 1980. For example, d04 has a coefficient of -8.766. This implies that the average of totfatrte was 8.8 less than 25.5 (average of totfatrte in 1980) or 16.8, precisely what is reported in

the graph above.

The fitted regression equation is avg fatality $= 25.49 + (-1.824) \cdot d81 + (-4.552) \cdot d82 + \cdots + (-8.766)$ · d04 All the estimated coefficients are statistically significant at 5%, except for the one for 1981, which suggests that there was a lot of variability in fatality rates across the 48 states in 1981. The intercept of the regression model is the average fatality rate for 1980. All other coefficients measure how the average fatality compares for the year (represented by the dummy variable) versus 1980. Each coefficient is negative, which means that the average fatality rate each year decreased relative to 1980. The coefficients are also mostly increasingly negative, represting the negative trend that we see in the graph above. Based on this, we would say that driving (as measured by fatality rate per 100K population) has gotten safer over the years 1980-2004 on average in the United States. Note, we have not accounted for any other factors in this assessment. The target variable totfatrte is a simple average of the fatality rates of each state, which is not weighted by the population of each state, nor for the cumulative miles driven.

## Expanded adding variables *bac08, bac10, perse, sbprim, sbsecon, sl70plus, gdl, perc14_24, unem, vehicmilespc*:

```
traffic$bac08bin = ifelse(traffic$bac08 < 0.5, 0, 1)
traffic$bac10bin = ifelse(traffic$bac10 <= 0.5, 0, 1)
traffic$persebin = ifelse(traffic$perse < 0.5, 0, 1)
traffic$sbprimbin = ifelse(traffic$sbprim < 0.5, 0, 1)
traffic$sbseconbin = ifelse(traffic$sbsecon < 0.5, 0, 1)
traffic$sl70plusbin = ifelse(traffic$sl70plus < 0.5, 0, 1)
traffic$gdlbin = ifelse(traffic$gdl < 0.5, 0, 1)

traffic$bac0810bin = traffic$bac08bin + traffic$bac10bin
describe(traffic$bac0810bin)
```

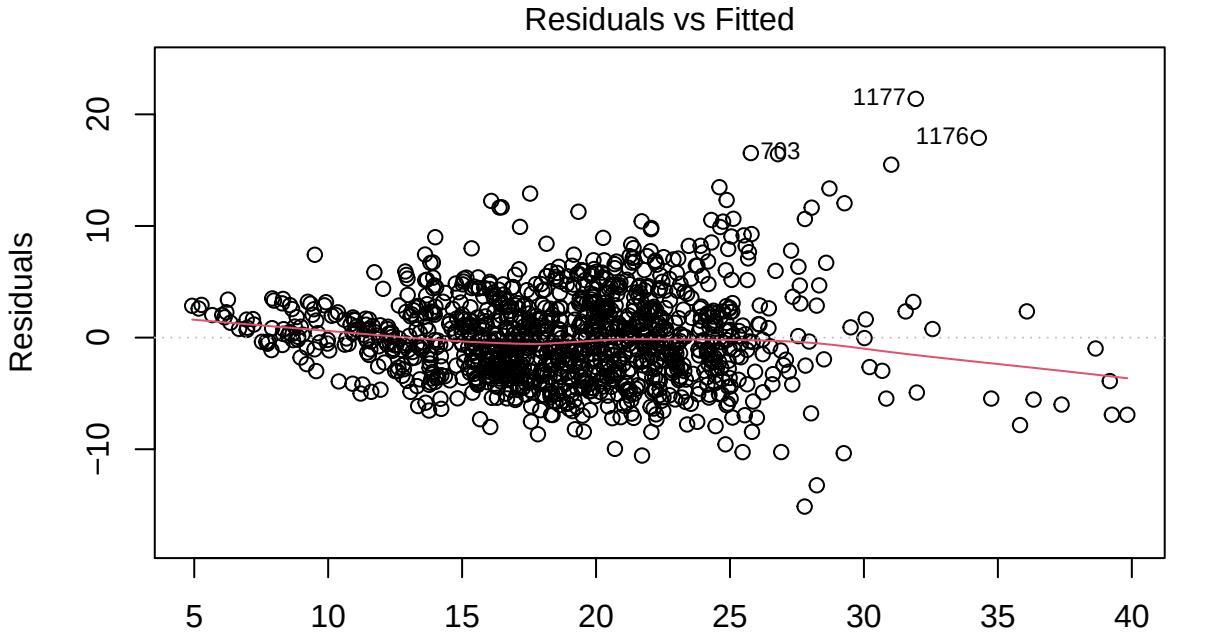
```
## traffic$bac0810bin
##        n  missing distinct     Info      Sum     Mean      Gmd
##     1200        0        2    0.417     1000   0.8333    0.278
```

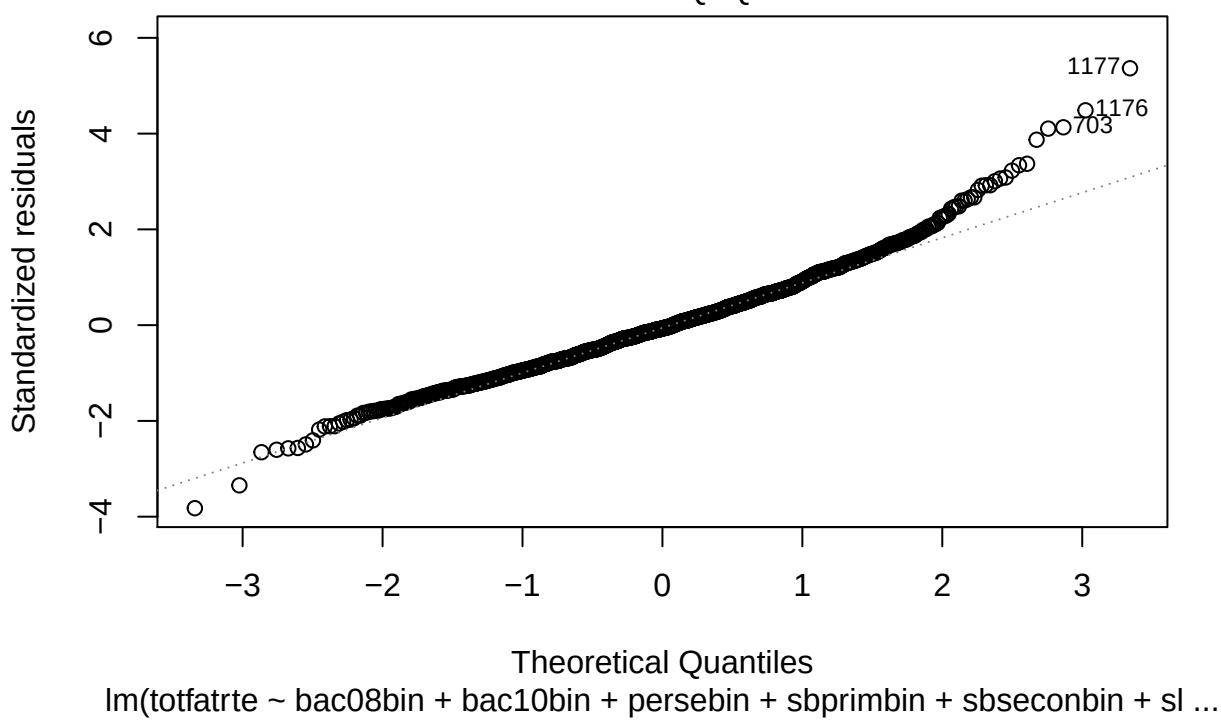
```
lm.mod2 = lm(totfatrte ~ bac08bin + bac10bin + persebin + sbprimbin + sbseconbin + sl70plusbin
    gdlbin + perc14_24 + unem + vehicmilespc + d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 +
    d89 + d90 + d91 + d92 + d93 + d94 + d95 + d96 + d97 + d98 + d99 + d00 + d01 + d02 + d03 +
    d04, data = traffic)
summary(lm.mod2)
```

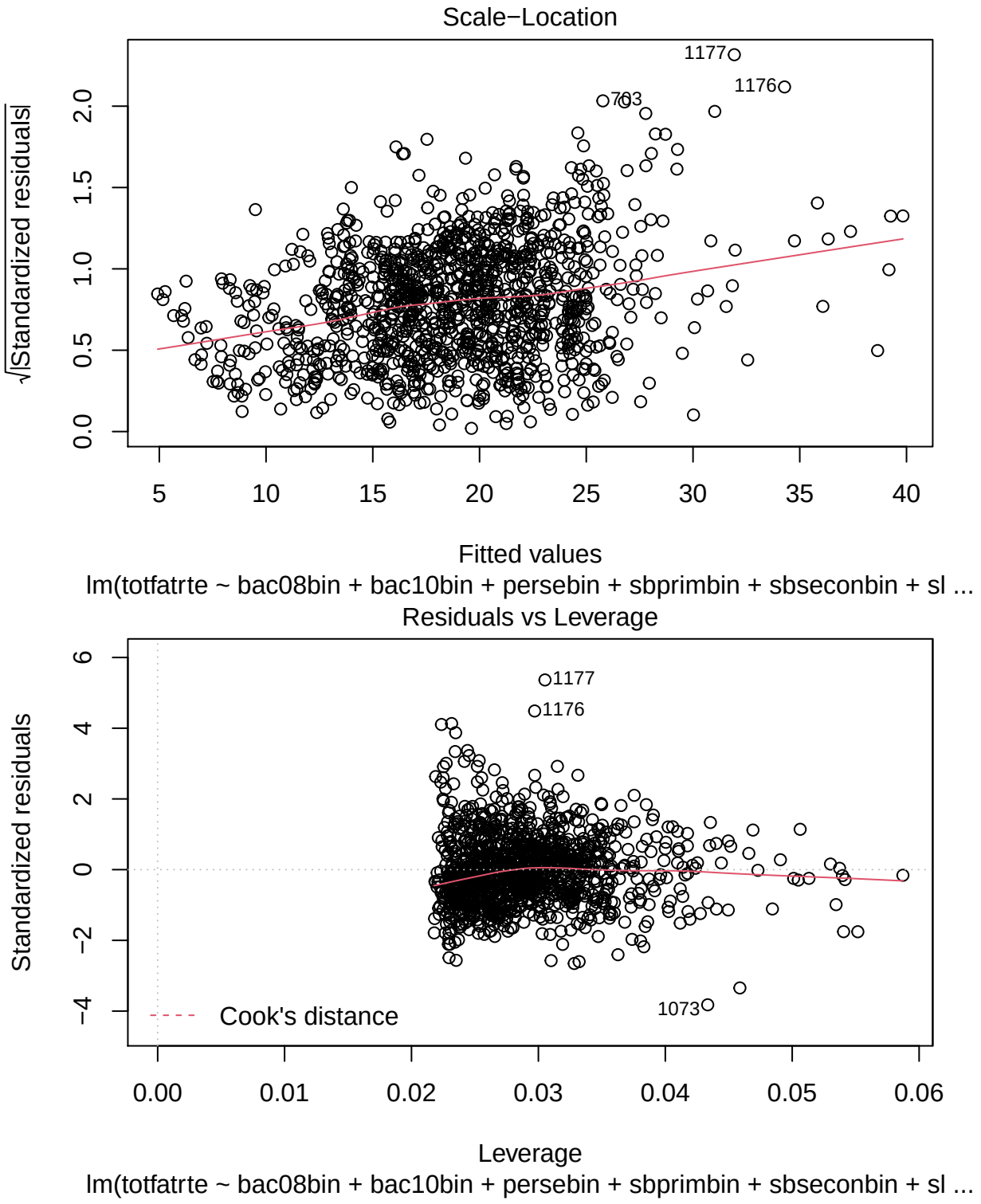
```
##
## Call:
## lm(formula = totfatrte ~ bac08bin + bac10bin + persebin + sbprimbin +
##     sbseconbin + sl70plusbin + gdlbin + perc14_24 + unem + vehicmilespc +
##     d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 + d89 + d90 +
##     d91 + d92 + d93 + d94 + d95 + d96 + d97 + d98 + d99 + d00 +
##     d01 + d02 + d03 + d04, data = traffic)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -15.137  -2.753  -0.272   2.305  21.385
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.95e+00   2.47e+00   -1.19   0.2330
## bac08bin     -2.63e+00   5.22e-01   -5.03  5.6e-07 ***
## bac10bin     -1.57e+00   3.85e-01   -4.07  5.1e-05 ***
## persebin     -5.74e-01   2.92e-01   -1.97   0.0496 *
## sbprimbin    -4.25e-02   4.91e-01   -0.09   0.9310
## sbseconbin    9.29e-02   4.29e-01    0.22   0.8287
## sl70plusbin   3.12e+00   4.34e-01    7.17  1.3e-12 ***
## gdlbin       -4.49e-01   5.06e-01   -0.89   0.3751
## perc14_24     1.50e-01   1.23e-01    1.22   0.2221
## unem          7.67e-01   7.78e-02    9.86  < 2e-16 ***
## vehicmilespc  2.93e-03   9.50e-05   30.87  < 2e-16 ***
## d81          -2.18e+00   8.28e-01   -2.63   0.0086 **
## d82          -6.61e+00   8.54e-01   -7.75  2.1e-14 ***
## d83          -7.47e+00   8.67e-01   -8.61  < 2e-16 ***
## d84          -5.80e+00   8.75e-01   -6.63  5.0e-11 ***
## d85          -6.43e+00   8.93e-01   -7.20  1.1e-12 ***
## d86          -5.79e+00   9.30e-01   -6.22  6.8e-10 ***
## d87          -6.30e+00   9.67e-01   -6.51  1.1e-10 ***
## d88          -6.52e+00   1.01e+00   -6.43  1.9e-10 ***
## d89          -7.99e+00   1.05e+00   -7.59  6.4e-14 ***
## d90          -8.88e+00   1.08e+00   -8.25  4.4e-16 ***
## d91          -1.10e+01   1.10e+00   -9.99  < 2e-16 ***
## d92          -1.28e+01   1.12e+00  -11.42  < 2e-16 ***
## d93          -1.27e+01   1.14e+00  -11.15  < 2e-16 ***
## d94          -1.23e+01   1.16e+00  -10.60  < 2e-16 ***
## d95          -1.19e+01   1.18e+00  -10.04  < 2e-16 ***
## d96          -1.39e+01   1.23e+00  -11.31  < 2e-16 ***
## d97          -1.40e+01   1.25e+00  -11.24  < 2e-16 ***
## d98          -1.48e+01   1.26e+00  -11.75  < 2e-16 ***
## d99          -1.49e+01   1.28e+00  -11.59  < 2e-16 ***
## d00          -1.52e+01   1.30e+00  -11.69  < 2e-16 ***
## d01          -1.59e+01   1.33e+00  -11.99  < 2e-16 ***
## d02          -1.65e+01   1.34e+00  -12.28  < 2e-16 ***
## d03          -1.68e+01   1.35e+00  -12.48  < 2e-16 ***
## d04          -1.65e+01   1.38e+00  -11.98  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.05 on 1165 degrees of freedom
## Multiple R-squared:  0.607,  Adjusted R-squared:  0.596
## F-statistic:   53 on 34 and 1165 DF,  p-value: <2e-16
```

```
plot(lm.mod2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(totfatrte ~ bac08bin + bac10bin + persebin + sbprimbin + sbseconbin + sl ...
1177
1176
703


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(totfatrte ~ bac08bin + bac10bin + persebin + sbprimbin + sbseconbin + sl ...
1177
1176
703




## Rationale In most instances we can expect there to be a lag between when certain types of laws are enacted and when they impact change in behavior. Given that, we suggest transforming the variables that are essentially binary in nature (whether a law was in effect or not) to be strictly binary (so if it was in effect for less than 1/2 the year, we will code as 0 and 1 otherwise).

As for the other variables - $perc14_24, unem, vehicmilespc$, they are all already expressed as normal-

ized metrics (where the denominator is some measure of the population, either as per 100 people or per person).

### Variable Definition

The BAC (blood alcohol content) is a measure used to determine if someone is driving under the influence of alcohol. For *bac*10 the threshold is 0.10 and *bac*8 represents a threshold of 0.08.

### Coefficient Interpretation

The coefficient on bac08 is -2.13 and the coefficient for bac10 is -1.12; The mathematical interpretation of the coefficients in the regression would suggest that all else being equal, a state that has a threshold of 0.08 would have a lower $totfatrte$ than a state that has a threshold of 0.1 because the coeff for bac08 is -2.13 and the coeff for bac10 is -1.12. This model structure also suggests that the effect of having neither of the thresholds implies an increase of the fatality rate by 3.25.

### Per Se Laws

According to the linear regression model - YES, perse laws have a negative effect on $totfatrte$. All else being equal, the existence of perse laws lowers $totfatrte$ by -0.57 compared to when these laws do not exist.

### Primary Seat Belt

Per the model specification, inclusion of the primary seatbelt variable *sbprimbin* implies a reduction in Fatalities by 0.0425. However, because the coefficient is not significantly different from zero, we interpret it as having no effect on $totfatrte$. This does not imply that seatbelts do not affect fatality rates, just that compulsory laws may not change drivers' willingness to comply with the law.

## Fixeed Effect Model

```
(effects.formula <- p.vars[-c(1)] %>%
    c(list(sep = "+")) %>%
    do.call(paste, .) %>%
    paste0(r.vars, "~", .) %>%
    as.formula)
```

```
## totfatrte ~ bac08 + bac10 + perse + sbprim + sbsecon + sl70plus +
##     gdl + perc14_24 + unem + vehicmilespc
## <environment: 0x7ff46b6af1d8>
```

```
(fe.plm <- plm(effects.formula, index = c("state", "year"), model = "within", data = data %>%
    mutate(sbprim = as.logical.factor(sbprim), sbsecon = as.logical.factor(sbsecon), bac08 = as
        bac10 = as.ternary.factor(bac10), perse = as.ternary.factor(perse), gdl = as.ternary.fa
        sl70plus = as.ternary.factor(sl70plus)))) %>%
    summary
```

```
## Oneway (individual) effect Within Model
##
```

```
## Call:
## plm(formula = effects.formula, data = data %>% mutate(sbprim = as.logical.factor(sbprim),
##     sbsecon = as.logical.factor(sbsecon), bac08 = as.ternary.factor(bac08),
##     bac10 = as.ternary.factor(bac10), perse = as.ternary.factor(perse),
##     gdl = as.ternary.factor(gdl), sl70plus = as.ternary.factor(sl70plus)),
##     model = "within", index = c("state", "year"))
##
## Balanced Panel: n = 48, T = 25, N = 1200
##
## Residuals:
##    Min. 1st Qu.  Median 3rd Qu.    Max.
## -7.4792 -1.1835 -0.0737  1.1185 14.3852
##
## Coefficients:
##               Estimate Std. Error t-value Pr(>|t|)
## bac081       -0.410888   0.605839   -0.68  0.49778
## bac082       -1.897288   0.383993   -4.94  8.9e-07 ***
## bac101       -0.765217   0.495467   -1.54  0.12276
## bac102       -1.451760   0.268656   -5.40  7.9e-08 ***
## perse1       -1.128947   0.465814   -2.42  0.01552 *
## perse2       -1.553706   0.248262   -6.26  5.5e-10 ***
## sbprimFALSE   1.809900   0.344934    5.25  1.8e-07 ***
## sbseconFALSE  0.862124   0.248095    3.47  0.00053 ***
## sl70plus1    -0.700472   0.443402   -1.58  0.11444
## sl70plus2    -1.165460   0.248038   -4.70  2.9e-06 ***
## gdl1         -1.272960   0.516080   -2.47  0.01379 *
## gdl2         -0.611440   0.231720   -2.64  0.00844 **
## perc14_24     0.943694   0.071088   13.28  < 2e-16 ***
## unem         -0.591677   0.051485  -11.49  < 2e-16 ***
## vehicmilespc  0.000295   0.000103    2.85  0.00441 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    12100
## Residual Sum of Squares: 5470
## R-Squared:      0.549
## Adj. R-Squared: 0.525
## F-statistic: 92.3251 on 15 and 1137 DF, p-value: <2e-16
```

The coefficients for bac08, bac10, perse and sbprim are all statistically significant with the panel fixed-effects model. The estimate for the bac08 coeff is slightly lower, the coeff for bac10 is about the same as before. The perse coeff has changed significanty from -6.1 to -1.1; We also fnid that the coeff for sbprim is now statistically significant.
Both models explain about the same amount of variation in the $totfatrte$ variable (comparable adjusted-R-sq values); however the fixed-effects model is more reliable because it models the variation over time in $totfatrte$ and all the independent variables *within* each state The fixed effect assumption is that the individual-specific effects are correlated with the independent variables. The pooled effects assumption (made in a random effects model) is that the individual-specific effects are

uncorrelated with the independent variables. This is a very strong assumption and a difficult one to meet. We are assuming there is no omitted variable bias and no correlation between same-state observations.

## Random Effects model versus Fixed Effects model:

```
(re.plm <- plm(effects.formula, index = c("state", "year"), model = "random", data = data %>%
    mutate(sbprim = as.logical.factor(sbprim), sbsecon = as.logical.factor(sbsecon), bac08 = as
        bac10 = as.ternary.factor(bac10), perse = as.ternary.factor(perse), gdl = as.ternary.fa
        sl70plus = as.ternary.factor(sl70plus)))) %>%
    summary
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = effects.formula, data = data %>% mutate(sbprim = as.logical.factor(sbprim),
##     sbsecon = as.logical.factor(sbsecon), bac08 = as.ternary.factor(bac08),
##     bac10 = as.ternary.factor(bac10), perse = as.ternary.factor(perse),
##     gdl = as.ternary.factor(gdl), sl70plus = as.ternary.factor(sl70plus)),
##     model = "random", index = c("state", "year"))
##
## Balanced Panel: n = 48, T = 25, N = 1200
##
## Effects:
##                 var std.dev share
## idiosyncratic 4.81    2.19  0.37
## individual    8.21    2.86  0.63
## theta: 0.849
##
## Residuals:
##    Min. 1st Qu.  Median 3rd Qu.    Max.
##  -6.101  -1.427  -0.247   1.035  16.665
##
## Coefficients:
##               Estimate Std. Error z-value Pr(>|z|)
## (Intercept)   2.120079   1.799874    1.18  0.23883
## bac081       -0.512775   0.634611   -0.81  0.41908
## bac082       -2.167240   0.397468   -5.45  5.0e-08 ***
## bac101       -0.906196   0.518827   -1.75  0.08070 .
## bac102       -1.553997   0.278986   -5.57  2.5e-08 ***
## perse1       -1.047119   0.487255   -2.15  0.03163 *
## perse2       -1.456766   0.255742   -5.70  1.2e-08 ***
## sbprimFALSE   1.893902   0.356310    5.32  1.1e-07 ***
## sbseconFALSE  0.982915   0.257904    3.81  0.00014 ***
## sl70plus1    -0.695566   0.464426   -1.50  0.13421
## sl70plus2    -1.185555   0.258051   -4.59  4.3e-06 ***
```

```
## gdl1          -1.310632   0.540980   -2.42  0.01541 *
## gdl2          -0.754621   0.242032   -3.12  0.00182 **
## perc14_24      1.006668   0.073148   13.76  < 2e-16 ***
## unem          -0.512683   0.053274   -9.62  < 2e-16 ***
## vehicmilespc   0.000543   0.000103    5.25  1.5e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    13000
## Residual Sum of Squares: 6270
## R-Squared:      0.517
## Adj. R-Squared: 0.511
## Chisq: 1267.01 on 15 DF, p-value: <2e-16
```

Random effects model would pool everything together and would rely on the assumption that same-state observations are independent. While we have quite a few variables we could add to the mix, this isn't a realistic assumption to make. We need to account for cultural/economic differences between states. We are able to remove this omitted variable bias using fixed effects modeling with dummy variables of each year to explain the unaccounted for time-variant error dependence.

## FE Estimated effect on *totfatrte* with increased traffic:

```
fe.plm$coefficients[["vehicmilespc"]] * 1000
```

```
## [1] 0.2948
```

```
re.plm$coefficients[["vehicmilespc"]] * 1000
```

```
## [1] 0.5434
```

Incrementing the per capita vehicle miles traveled by 1,000 leads to an increase in total fatality rate per 100,000 people by .295 according to our fixed effects model and .543 according to our random effects model - both of which are controlling for year and state.

#Serial correlation or heteroskedasticity in the idiosyncratic errors of the model: We will use the Breusch-Godfrey/Wooldridge test for serial correlation:

```
plm::pbgtest(fe.plm)
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  effects.formula
## chisq = 395, df = 25, p-value <2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

```
plm::pbgtest(re.plm)
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
```

```
## data:  effects.formula
## chisq = 441, df = 25, p-value <2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

Observing the above outputs, we can see that both our random effects and fixed effects model reject the null hypothesis that there is no serial correlation between our residuals. Next, we will test the null hypothesis of homoskedastic residuals using the Breusch-Pagan test:

```
library("lmtest")
lmtest::bptest(fe.plm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fe.plm
## BP = 96, df = 15, p-value = 8e-14
```

```
lmtest::bptest(re.plm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  re.plm
## BP = 96, df = 15, p-value = 8e-14
```

Observing the above outputs, both of our models reject the null hypothesis of homoskedastic residuals. To control heteroskedastic residuals and serial correlation, we can use Robust Covariance Matrix Estimation (Sandwich Estimator) with the "arellano" method, which clusters by group and is best with models showing heteroskedasticity/serial correlation:

```
coeftest(fe.plm, vcovHC(fe.plm, method = "arellano"))
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value Pr(>|t|)
## bac081       -0.410888   0.621093   -0.66  0.50839
## bac082       -1.897288   0.660819   -2.87  0.00417 **
## bac101       -0.765217   0.449456   -1.70  0.08893 .
## bac102       -1.451760   0.495162   -2.93  0.00344 **
## perse1       -1.128947   0.458036   -2.46  0.01386 *
## perse2       -1.553706   0.450420   -3.45  0.00058 ***
## sbprimFALSE   1.809900   0.745551    2.43  0.01535 *
## sbseconFALSE  0.862124   0.473970    1.82  0.06918 .
## sl70plus1    -0.700472   0.389479   -1.80  0.07237 .
## sl70plus2    -1.165460   0.553906   -2.10  0.03559 *
## gdl1         -1.272960   0.557631   -2.28  0.02263 *
## gdl2         -0.611440   0.337415   -1.81  0.07023 .
## perc14_24     0.943694   0.170636    5.53    4e-08 ***
## unem         -0.591677   0.078593   -7.53    1e-13 ***
## vehicmilespc  0.000295   0.000271    1.09  0.27661
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(re.plm, vcovHC(re.plm, method = "arellano"))
```

```
##
## t test of coefficients:
##
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.120079   4.154475    0.51  0.60993
## bac081       -0.512775   0.619932   -0.83  0.40832
## bac082       -2.167240   0.682350   -3.18  0.00153 **
## bac101       -0.906196   0.473738   -1.91  0.05601 .
## bac102       -1.553997   0.514951   -3.02  0.00260 **
## perse1       -1.047119   0.446017   -2.35  0.01905 *
## perse2       -1.456766   0.429919   -3.39  0.00073 ***
## sbprimFALSE   1.893902   0.729083    2.60  0.00950 **
## sbseconFALSE  0.982915   0.478573    2.05  0.04021 *
## sl70plus1    -0.695566   0.379306   -1.83  0.06694 .
## sl70plus2    -1.185555   0.529922   -2.24  0.02546 *
## gdl1         -1.310632   0.555105   -2.36  0.01838 *
## gdl2         -0.754621   0.334390   -2.26  0.02421 *
## perc14_24     1.006668   0.168779    5.96  3.2e-09 ***
## unem         -0.512683   0.079653   -6.44  1.8e-10 ***
## vehicmilespc  0.000543   0.000249    2.18  0.02953 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Because the panel data is comparing time series, we can expect to see serial correlation and heteroskedasticity, so using the robust errors is necessary. The consequences of not using heteroskedasticity robust standard errors is that we would underestimate our standard errors, thus falsely inflating the significance of each of our reported coefficients. We observe less significance when using robust standard errors above - though our key findings remain.