Statistical Methods for Discrete Response, Time Series, and Panel Data (W271): Lab 4

K.C. Tobin, Weixing Sun, Winston Lin

August 19, 2017

Description of the Lab

In this lab, you are asked to answer the question **"Do changes in traffic laws affect traffic fatalities?"** To do so, you will conduct the tasks specified below using the data set *driving.Rdata*, which includes 25 years of data that cover changes in various state drunk driving, seat belt, and speed limit laws.

- Specifically, this data set contains data for the 48 continental U.S. states from 1980 through 2004. Various driving laws are indicated in the data set, such as the alcohol level at which drivers are considered legally intoxicated. There are also indicators for "per se" laws—where licenses can be revoked without a trial—and seat belt laws. A few economics and demographic variables are also included. The description of the each of the variables in the dataset is come with the dataste.

Exercises:

1. Load the data. Provide a description of the basic structure of the dataset, as we have done in throughout the semester.

- The panel data set consists of 56 variables with 1200 observations. The unit of analysis are the 48 continental states of the U.S., which are coded as a single numeric variable, sorted alphabetically. The data are measured from 1980 to 2004, and the years are coded as a numeric single variable as well as a full set of dummy variables. This is a balanced panel data set since there are observations for all 48 states for every year.
- The dependent variables are fatality counts and ratios of those counts to distance and population. The independent variables include continuous variables for state demographics, such as population, unemployment rate, and percentage of population ages 18-24, as well as driving activity, such as speed limits and miles driven per capita. The independent variables also consist of indicator variables that represent driving laws related to requirements for drivers licenses (GDL), seat belt usage, blood alcohol content (BAC) limits, and penalties for drunk driving. All indicator and dummy variables are binary.

```
library(car)
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

```
library(ggplot2)
library(GGally)
library(plm)
```

```
## Loading required package: Formula
```

```
# load data set and view structure
load('driving.RData')
```

```r
attach(data)

# preview and summary stats
str(data)
```

```
## 'data.frame':    1200 obs. of  56 variables:
##  $ year        : int  1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 ...
##  $ state       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ sl55        : num  1 1 1 1 1 ...
##  $ sl65        : num  0 0 0 0 0 ...
##  $ sl70        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ sl75        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ slnone      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ seatbelt    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ minage      : num  18 18 18 18 18 20 21 21 21 21 ...
##  $ zerotol     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ gdl         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bac10       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ bac08       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ perse       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ totfat      : int  940 933 839 930 932 882 1080 1111 1024 1029 ...
##  $ nghtfat     : int  422 434 376 397 421 358 500 499 423 418 ...
##  $ wkndfat     : int  236 248 224 223 237 224 279 300 226 247 ...
##  $ totfatpvm   : num  3.2 3.35 2.81 3 2.83 ...
##  $ nghtfatpvm  : num  1.44 1.56 1.26 1.28 1.28 ...
##  $ wkndfatpvm  : num  0.803 0.89 0.75 0.719 0.72 ...
##  $ statepop    : int  3893888 3918520 3925218 3934109 3951834 3972527 3991569 4015261 4023858 403022
##  $ totfatrte   : num  24.1 24.1 21.4 23.6 23.6 ...
##  $ nghtfatrte  : num  10.84 11.08 9.58 10.09 10.65 ...
##  $ wkndfatrte  : num  6.06 6.33 5.71 5.67 6 ...
##  $ vehicmiles  : num  29.4 27.9 29.9 31 32.9 ...
##  $ unem        : num  8.8 10.7 14.4 13.7 11.1 ...
##  $ perc14_24   : num  18.9 18.7 18.4 18 17.6 ...
##  $ sl70plus    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ sbprim      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sbsecon     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d80         : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ d81         : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ d82         : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ d83         : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ d84         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ d85         : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ d86         : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ d87         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ d88         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ d89         : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ d90         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d91         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d92         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d93         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d94         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d95         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d96         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d97         : int  0 0 0 0 0 0 0 0 0 0 ...
```

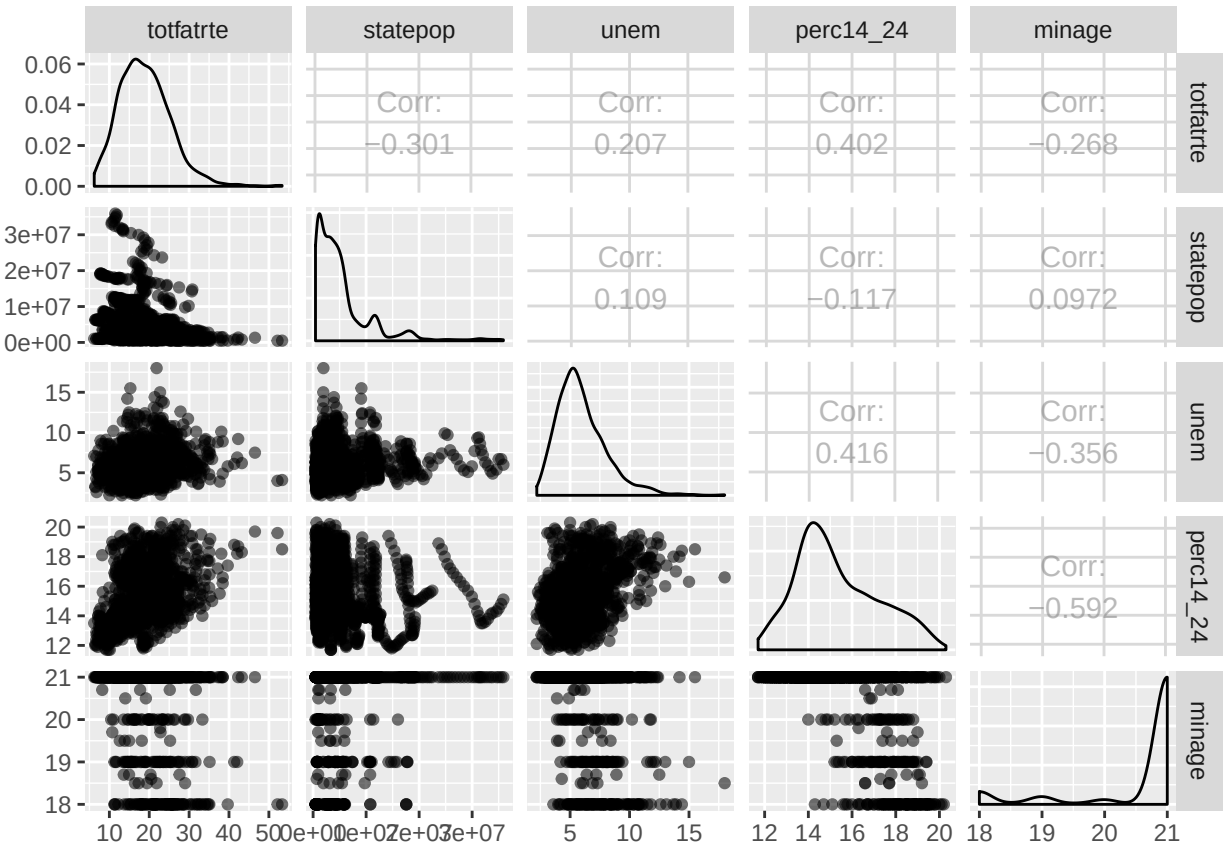
```
##  $ d98         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d99         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d00         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d01         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d02         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d03         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ d04         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ vehicmilespc: num  7544 7108 7607 7880 8334 ...
##  - attr(*, "datalabel")= chr ""
##  - attr(*, "time.stamp")= chr "22 Jan 2013 14:09"
##  - attr(*, "formats")= chr  "%8.0g" "%8.0g" "%9.0g" "%9.0g" ...
##  - attr(*, "types")= int  252 251 254 254 254 254 254 251 254 254 ...
##  - attr(*, "val.labels")= chr  "" "" "" "" ...
##  - attr(*, "var.labels")= chr  "1980 through 2004" "48 continental states, alphabetical" "speed limi
##  - attr(*, "version")= int 12
```

```
# head(data)
# summary(data)
```

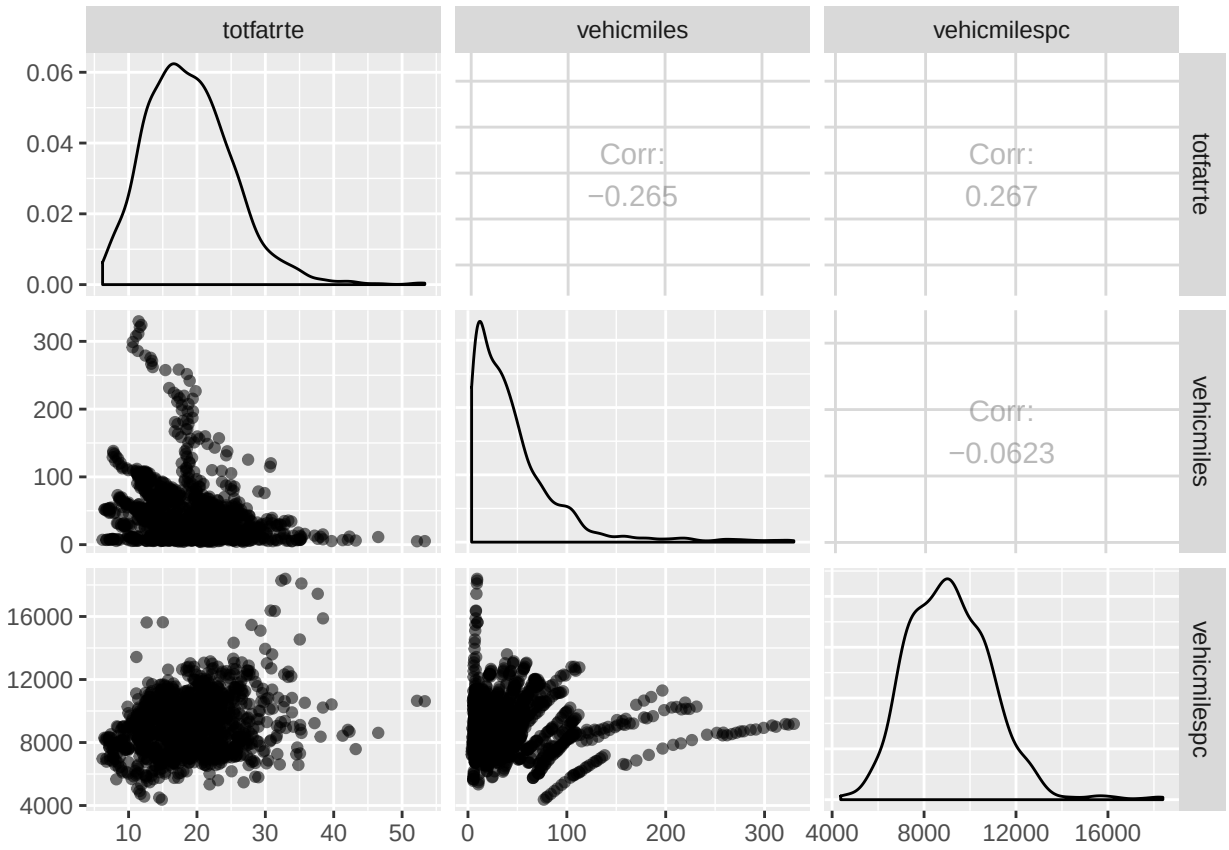
Conduct a very thorough EDA, which should include both graphical and tabular techniques, on the dataset, including both the dependent variable *totfatrte* and the potential explanatory variables. Remember, graphs must be well-labeled. You need to write a detailed narrative of your observations of your EDA. *Reminder: giving an "output dump" (i.e. providing a bunch of graphs and tables without description and hoping your audience will interpret them) will receive zero point in this exercise.*

- Histograms of the continuous variables are right skewed to varying degrees, which suggests we may want to apply a log transform to these variables to satisfy the normality assumption for OLS inference.
- Scatterplots show positive correlations between fatality rates and unemployment rates, percentage of population ages 14-24, and miles driven per capita, as well as negative correlations between fatality rates and state population, minimum drinking age, and miles driven.

```
### EDA for continuous variables - (need to examine for each year)

# scatterplots
ggpairs(data[c('totfatrte',  # slightly skewed right --> log transform
               'statepop',  # heavily skewed right --> log transform; low neg. correlation w/ fat. rate
               'unem',  # skewed right --> log transform; low pos. correlation w/ fat. rate
               'perc14_24',  # slightly skewed right --> log transform; pos. correlation w/ fat. rate
               'minage')],  # heavily skewed left --> log transform; low neg. correlation w/ fat. rate
        aes(alpha=0.1))
```




```
# scatterplots
ggpairs(data[c('totfatrte',  # slightly skewed right --> log transform
               'vehicmiles',  # heavily skewed right --> log transform; low neg. correlation w/ fat. ra
               'vehicmilespc'  # slightly skewed right --> log transform; low pos. correlation w/ fat.
               )],  # --> transform to binary
        aes(alpha=0.1))
```



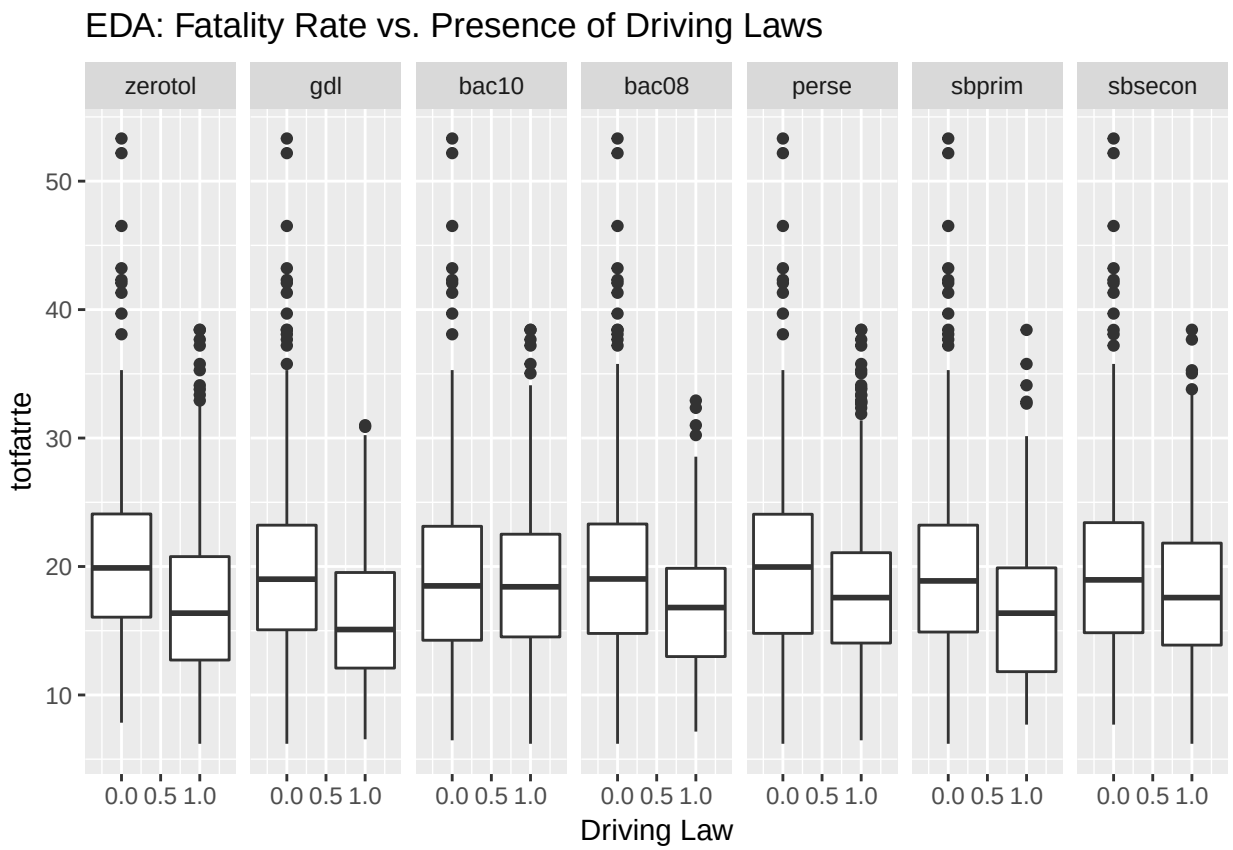
- Boxplots of indicator variables across driving laws show lower median and max fatality rate for state-years that enacted stricter seat belt usage, graduated drivers license, zero tolerance, per se, and BAC laws. Median and max fatality rates appear to increase for states with freeway speed limits greater than 70mph, although the opposite is true for states with speed limit of 55mph.
- Boxplots of the indicator variables across states show that the median fatality rates differ significantly from each other and may
- These correlations may be helpful for checking our intuitions about how the explanatory variables should affect fatality rates. However, the plots used are based on pooled data and do not account for the fact that the data are from multiple time periods.

```
### EDA for indicator variables - (need to examine for each year)

# reshape data for facet grid
require(reshape2)
```

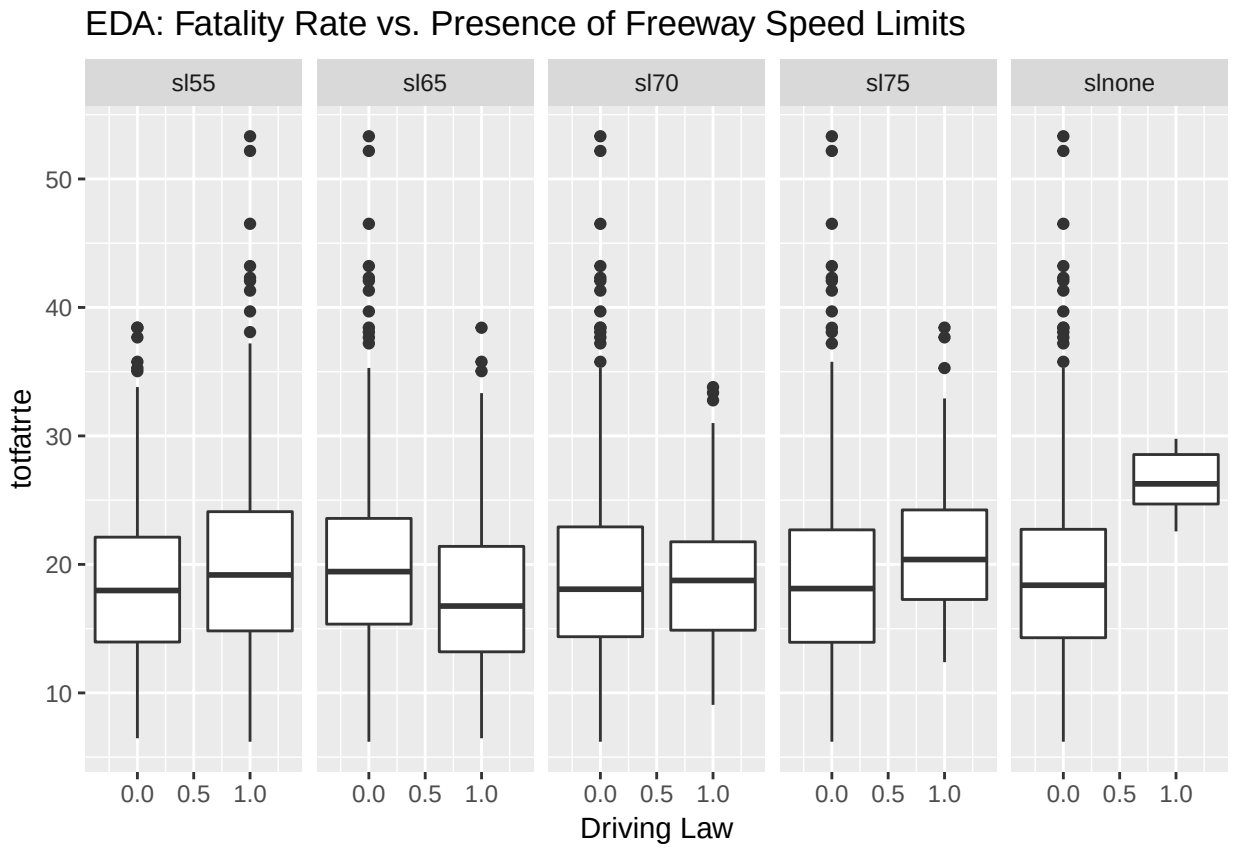
```
## Loading required package: reshape2
```

```
df.melt = melt(data[c('state',
                      'year',
                      'totfatrte',
                      'zerotol',  # =1 --> lower median/max fat. rate
                      'gdl',  # =1 --> lower median/max fat. rate
                      'bac10',  # =1 --> no change in median fat. rate
                      'bac08',  # =1 --> lower median/max fat. rate
                      'perse',  # =1 --> lower median/max fat. rate
                      'sbprim',  # =1 --> lower median/max fat. rate
                      'sbsecon')],  # =1 --> lower median/max fat. rate
```

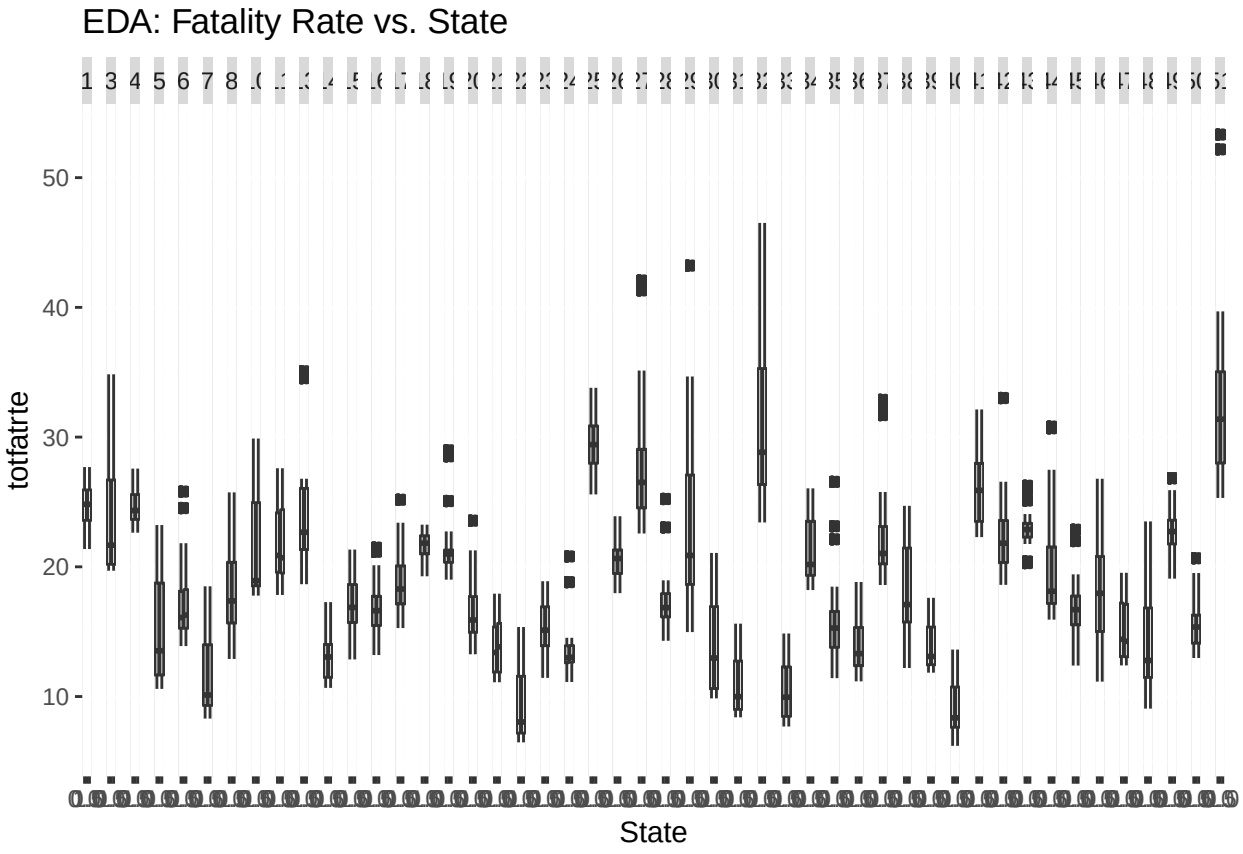
```
                id.vars=c('state','year','totfatrte'))

# boxplots by driving laws
plot1 = ggplot(data=df.melt, aes(x=round(value), y=totfatrte)) + geom_boxplot(aes(group=round(value))) +
plot1 + labs(title='EDA: Fatality Rate vs. Presence of Driving Laws', x='Driving Law', y='totfatrte')
```




```
# reshape data for facet grid
df.melt = melt(data[c('state',
                      'year',
                      'totfatrte',
                      'sl55',  # =1 --> higher median/max fat. rate; nearly binary --> transform to bin
                      'sl65',  # =1 --> lower median/max fat. rate; nearly binary --> transform to bina
                      'sl70',  # =1 --> lower median fat. rate; nearly binary --> transform to binary?
                      'sl75',  # =1 --> higher median fat. rate; nearly binary --> transform to binary?
                      'slnone')],  # =1 --> higher median fat. rate; nearly binary --> transform to bin
                id.vars=c('state','year','totfatrte'))

# boxplots by speed limit
plot2 = ggplot(data=df.melt, aes(x=round(value), y=totfatrte)) + geom_boxplot(aes(group=round(value))) +
plot2 + labs(title='EDA: Fatality Rate vs. Presence of Freeway Speed Limits', x='Driving Law', y='totfa
```




```
# boxplots by state - not sure how to tag state IDs with names
plot3 = ggplot(data=df.melt, aes(x=round(value), y=totfatrte)) + geom_boxplot(aes(group=round(value))) +
plot3 + labs(title='EDA: Fatality Rate vs. State', x='State', y='totfatrte')
```



2. How is the our dependent variable of interest *totfatrte* defined? What is the average of this variable in each of the years in the time period covered in this dataset?

- totfatrte is the total number of fatalities per 100,000 population. As we can see in the plot the average fatalities decreases over the time period. Our pooled model estimates this exact average with a baseline year and differences from the baseline. Driving safety as measured in total number of fatalities per 100,000 population certainly decreased as to whether or not this constitutes an overall increase in safety would require further analysis. The average fatality rates for 1980-2004 are below:

```
round(aggregate(totfatrte ~ year, FUN=mean, data=data), 2)
```

```
##    year totfatrte
## 1  1980     25.49
## 2  1981     23.67
## 3  1982     20.94
## 4  1983     20.15
## 5  1984     20.27
## 6  1985     19.85
## 7  1986     20.80
## 8  1987     20.77
## 9  1988     20.89
## 10 1989     19.77
## 11 1990     19.51
## 12 1991     18.09
## 13 1992     17.16
## 14 1993     17.13
## 15 1994     17.16
```

```
## 16 1995     17.67
## 17 1996     17.37
## 18 1997     17.61
## 19 1998     17.27
## 20 1999     17.25
## 21 2000     16.83
## 22 2001     16.79
## 23 2002     17.03
## 24 2003     16.76
## 25 2004     16.73
```

Estimate a very simple regression model of totfatrte on dummy variables for the years 1981 through 2004. What does this model explain? Describe what you find in this model. Did driving become safer over this period? Please provide a detailed explanation.

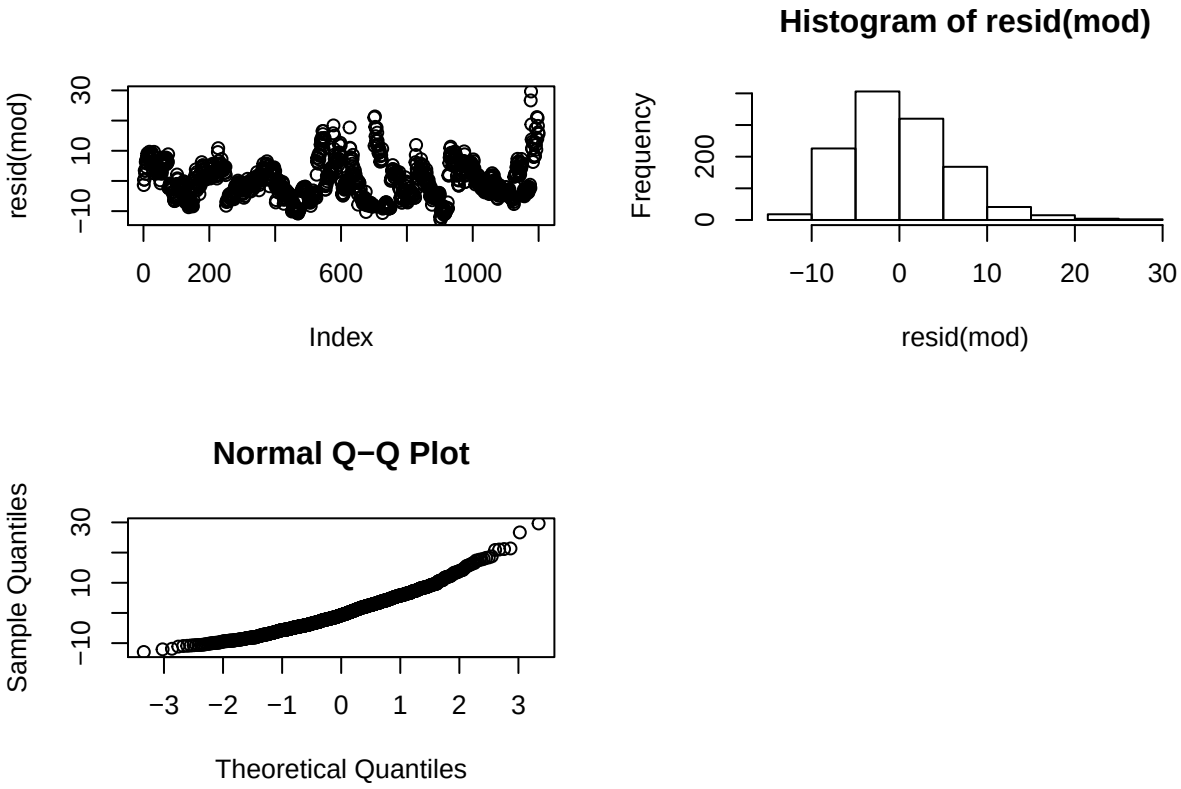
- This model explains the change in fatality rate in each year relative to 1980. The coefficients for each year dummy variable can be interpreted as the approximate change in percentage points in the average fatality rate taken across all 48 states. All coefficients are negative and highly statistically significant (at $\alpha = 0.001$). The residuals reveal some heteroskedasticity and non-normality, confirming that we need to transform the response to satisfy assumptions for OLS and make our inferences valid.
- From 1980 to 2004, the average fatality rate decreased by about 8.8 points, which matches the decrease when we manually calculate from the averages above ($25.49 - 16.73$). Using the average fatality rate as a measure of driving safety, we could claim that the decrease in the fatality rate form 1980 levels implies that driving became safer in 2004.

```
# panel data frame
df.plm = pdata.frame(data, index = c('state','year'), drop.index = F)

# model formula
form0 = totfatrte ~ d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 + d89 +
                    d90 + d91 + d92 + d93 + d94 + d95 + d96 + d97 + d98 + d99 +
                    d00 + d01 + d02 + d03 + d04

# fit model - pooled OLS ~ year dummy vars, 1980 as base year
mod0 = plm(form0, data = df.plm, model='pooling')
summary(mod0)
```

```
## Pooling Model
##
## Call:
## plm(formula = form0, data = df.plm, model = "pooling")
##
## Balanced Panel: n=48, T=25, N=1200
##
## Residuals :
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -12.93021  -4.34682  -0.73052   3.74875  29.64979
##
## Coefficients :
##             Estimate Std. Error t-value  Pr(>|t|)
## (Intercept) 25.49458    0.86712 29.4015 < 2.2e-16 ***
## d81         -1.82438    1.22629 -1.4877 0.1370936
## d82         -4.55208    1.22629 -3.7121 0.0002152 ***
## d83         -5.34167    1.22629 -4.3560 1.440e-05 ***
```

```
## d84         -5.22708    1.22629 -4.2625 2.183e-05 ***
## d85         -5.64313    1.22629 -4.6018 4.644e-06 ***
## d86         -4.69417    1.22629 -3.8279 0.0001360 ***
## d87         -4.71979    1.22629 -3.8488 0.0001251 ***
## d88         -4.60292    1.22629 -3.7535 0.0001829 ***
## d89         -5.72229    1.22629 -4.6663 3.418e-06 ***
## d90         -5.98938    1.22629 -4.8841 1.182e-06 ***
## d91         -7.39979    1.22629 -6.0343 2.137e-09 ***
## d92         -8.33667    1.22629 -6.7983 1.681e-11 ***
## d93         -8.36688    1.22629 -6.8229 1.425e-11 ***
## d94         -8.33938    1.22629 -6.8005 1.656e-11 ***
## d95         -7.82604    1.22629 -6.3819 2.512e-10 ***
## d96         -8.12521    1.22629 -6.6258 5.246e-11 ***
## d97         -7.88396    1.22629 -6.4291 1.863e-10 ***
## d98         -8.22917    1.22629 -6.7106 3.007e-11 ***
## d99         -8.24417    1.22629 -6.7228 2.774e-11 ***
## d00         -8.66896    1.22629 -7.0692 2.666e-12 ***
## d01         -8.70188    1.22629 -7.0961 2.214e-12 ***
## d02         -8.46500    1.22629 -6.9029 8.316e-12 ***
## d03         -8.73104    1.22629 -7.1199 1.877e-12 ***
## d04         -8.76563    1.22629 -7.1481 1.542e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    48612
## Residual Sum of Squares: 42407
## R-Squared:      0.12765
## Adj. R-Squared: 0.10983
## F-statistic: 7.16387 on 24 and 1175 DF, p-value: < 2.22e-16
```

```r
# check residuals
mod_diag = function(mod) {
        par(mfrow=c(2,2))
        plot(resid(mod))
        hist(resid(mod))
        qqnorm(resid(mod))
}

mod_diag(mod0)
```



3. Expand your model in *Exercise 2* by adding variables *bac08, bac10, perse, sbprim, sbsecon, sl70plus, gdl, perc14_24, unem, vehicmilespc*, and perhaps *transformations of some or all of these variables*. Please explain carefully your rationale, which should be based on your EDA, behind any transformation you made. If no transformation is made, explain why transformation is not needed.

- In our EDA, histograms of the continuous variables totfatrte, statepop, unem, perc14_24, and vehicmilespc showed significant skew, so we applied log transformations to enforce normality on these variables and make our inference valid. This also stabilizes the variance, as seen in the residual plots.
- Indicator variables for bac08, bac10, perse, sbprim, sbsecon, sl70plus, and gdl were transformed to binary variables to make interpretation easier. They represent fractional years to account for laws changing in the middle of the year, which is less intuitive than simply indicating whether a state had a law or not in a given year in order to determine if these laws affected fatality rates. The model with the binarized variables has similar estimates for the coefficients, so we could've used the original variables to fit the model as well if we wanted to look at differences in effect between whole and parts of a year.
- The residuals appear to be normally distributed, but not homoskedastic. They cluster at regular intervals when plotted, indicating that there could be an AR process in the data. We would expect this from a pooled OLS model though, which doesn't take time into account.

```
# log transforms - enforce normality
df.plm$ltotfatrte = log(df.plm$totfatrte)
df.plm$lunem = log(df.plm$unem)
df.plm$lperc14_24 = log(df.plm$perc14_24)
df.plm$lvehicmilespc = log(df.plm$vehicmilespc)

# binarize law variables
df.plm$bac08.round <- factor(round(df.plm$bac08))
df.plm$bac10.round <-factor(round(df.plm$bac10))
```

```r
df.plm$perse.round <-factor(round(df.plm$perse))
df.plm$sbprim.round <-factor(round(df.plm$sbprim))
df.plm$sbsecon.round <-factor(round(df.plm$sbsecon))
df.plm$s170plus.round <-factor(round(df.plm$sl70plus))
df.plm$gdl.round <- factor(round(df.plm$gdl))

# model formula
form1 = update(form0, ltotfatrte ~ . + bac08 + bac10 + perse + sbprim + sbsecon + sl70plus + gdl +
                       lperc14_24 + lunem + lvehicmilespc)

# fit model - pooled OLS ~ transformed dependent/independent vars
mod1 = plm(form1, data=df.plm, model='pooling')
summary(mod1)
```

```
## Pooling Model
##
## Call:
## plm(formula = form1, data = df.plm, model = "pooling")
##
## Balanced Panel: n=48, T=25, N=1200
##
## Residuals :
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -0.58140186 -0.12464785 -0.00088522  0.14026490  0.62183160
##
## Coefficients :
##                  Estimate  Std. Error  t-value  Pr(>|t|)
## (Intercept)   -1.1717e+01  4.4470e-01 -26.3476 < 2.2e-16 ***
## d81           -9.2448e-02  4.1062e-02  -2.2514  0.024544 *
## d82           -2.9551e-01  4.1863e-02  -7.0590 2.875e-12 ***
## d83           -3.4987e-01  4.2534e-02  -8.2256 5.145e-16 ***
## d84           -3.0010e-01  4.3240e-02  -6.9403 6.481e-12 ***
## d85           -3.3772e-01  4.4076e-02  -7.6622 3.830e-14 ***
## d86           -3.1420e-01  4.5833e-02  -6.8552 1.152e-11 ***
## d87           -3.4997e-01  4.7675e-02  -7.3408 3.975e-13 ***
## d88           -3.6001e-01  5.0079e-02  -7.1889 1.165e-12 ***
## d89           -4.4473e-01  5.1984e-02  -8.5552 < 2.2e-16 ***
## d90           -5.0361e-01  5.3133e-02  -9.4783 < 2.2e-16 ***
## d91           -6.1742e-01  5.4301e-02 -11.3702 < 2.2e-16 ***
## d92           -7.2308e-01  5.5410e-02 -13.0498 < 2.2e-16 ***
## d93           -7.1367e-01  5.6156e-02 -12.7088 < 2.2e-16 ***
## d94           -7.0037e-01  5.7295e-02 -12.2239 < 2.2e-16 ***
## d95           -6.7953e-01  5.8738e-02 -11.5689 < 2.2e-16 ***
## d96           -8.0021e-01  6.0845e-02 -13.1515 < 2.2e-16 ***
## d97           -8.2008e-01  6.2348e-02 -13.1534 < 2.2e-16 ***
## d98           -8.6319e-01  6.3256e-02 -13.6460 < 2.2e-16 ***
## d99           -8.6437e-01  6.4263e-02 -13.4505 < 2.2e-16 ***
## d00           -8.7656e-01  6.5363e-02 -13.4106 < 2.2e-16 ***
## d01           -9.2754e-01  6.6200e-02 -14.0112 < 2.2e-16 ***
## d02           -9.7180e-01  6.6598e-02 -14.5920 < 2.2e-16 ***
## d03           -9.9245e-01  6.7112e-02 -14.7879 < 2.2e-16 ***
## d04           -9.7850e-01  6.8551e-02 -14.2740 < 2.2e-16 ***
## bac08         -6.3373e-02  2.6690e-02  -2.3744  0.017739 *
## bac10         -1.8650e-02  1.9674e-02  -0.9479  0.343355
```

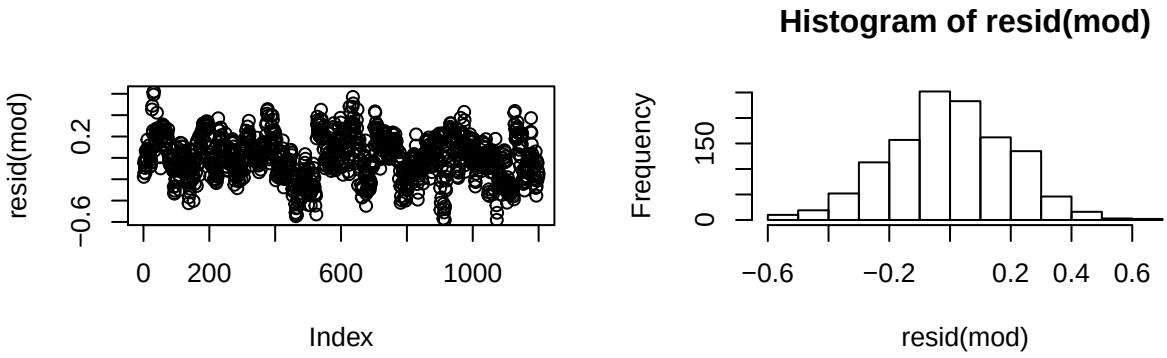
```
## perse         -1.9890e-02  1.4811e-02  -1.3429  0.179578
## sbprim         5.5852e-05  2.4521e-02   0.0023  0.998183
## sbsecon        2.0230e-02  2.1380e-02   0.9462  0.344238
## sl70plus       2.2965e-01  2.2186e-02  10.3513 < 2.2e-16 ***
## gdl           -2.6547e-02  2.6077e-02  -1.0180  0.308873
## lperc14_24     2.8841e-01  9.2747e-02   3.1096  0.001919 **
## lunem          2.6421e-01  2.4087e-02  10.9690 < 2.2e-16 ***
## lvehicmilespc  1.5316e+00  4.4497e-02  34.4198 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    142.38
## Residual Sum of Squares: 47.013
## R-Squared:      0.6698
## Adj. R-Squared: 0.66017
## F-statistic: 69.5063 on 34 and 1165 DF, p-value: < 2.22e-16
```

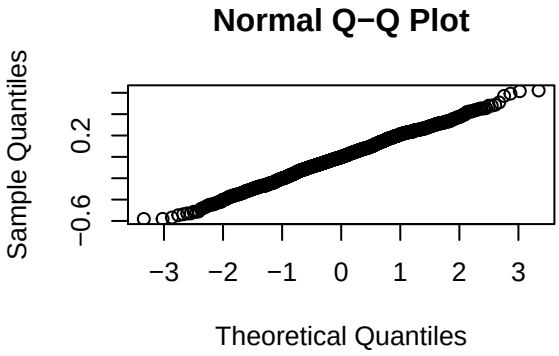
```
# check residuals - not white noise, but normal
mod_diag(mod1)

# fit model - pooled OLS ~ binarized law vars
form1b = update(form0, ltotfatrte ~ . + bac08.round + bac10.round + perse.round + sbprim.round + sbseco
                       s170plus.round + gdl.round + lperc14_24 + lunem + lvehicmilespc)
mod1b <- plm(form1b, data = df.plm, model = "pooling")
summary(mod1b)
```

```
## Pooling Model
##
## Call:
## plm(formula = form1b, data = df.plm, model = "pooling")
##
## Balanced Panel: n=48, T=25, N=1200
##
## Residuals :
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -0.58519646 -0.12671734 -0.00098578  0.14101572  0.62265979
##
## Coefficients :
##                    Estimate  Std. Error  t-value  Pr(>|t|)
## (Intercept)     -1.1771e+01  4.4431e-01 -26.4933 < 2.2e-16 ***
## d81             -9.2627e-02  4.1162e-02  -2.2503  0.024616 *
## d82             -2.9696e-01  4.1957e-02  -7.0777 2.526e-12 ***
## d83             -3.5379e-01  4.2471e-02  -8.3303 2.243e-16 ***
## d84             -3.0170e-01  4.3104e-02  -6.9994 4.331e-12 ***
## d85             -3.3910e-01  4.3937e-02  -7.7180 2.530e-14 ***
## d86             -3.1513e-01  4.5779e-02  -6.8838 9.502e-12 ***
## d87             -3.5064e-01  4.7630e-02  -7.3617 3.423e-13 ***
## d88             -3.6094e-01  5.0054e-02  -7.2110 9.968e-13 ***
## d89             -4.4504e-01  5.1963e-02  -8.5645 < 2.2e-16 ***
## d90             -5.0443e-01  5.3102e-02  -9.4994 < 2.2e-16 ***
## d91             -6.1921e-01  5.4227e-02 -11.4189 < 2.2e-16 ***
## d92             -7.2449e-01  5.5371e-02 -13.0843 < 2.2e-16 ***
## d93             -7.1591e-01  5.6073e-02 -12.7676 < 2.2e-16 ***
## d94             -7.0260e-01  5.7174e-02 -12.2888 < 2.2e-16 ***
## d95             -6.7845e-01  5.8704e-02 -11.5572 < 2.2e-16 ***
```

```
## d96             -8.0304e-01  6.0898e-02 -13.1865 < 2.2e-16 ***
## d97             -8.1178e-01  6.1958e-02 -13.1021 < 2.2e-16 ***
## d98             -8.6016e-01  6.3108e-02 -13.6299 < 2.2e-16 ***
## d99             -8.6217e-01  6.3881e-02 -13.4964 < 2.2e-16 ***
## d00             -8.7475e-01  6.4925e-02 -13.4732 < 2.2e-16 ***
## d01             -9.3055e-01  6.5327e-02 -14.2445 < 2.2e-16 ***
## d02             -9.7524e-01  6.5749e-02 -14.8328 < 2.2e-16 ***
## d03             -9.9868e-01  6.5988e-02 -15.1342 < 2.2e-16 ***
## d04             -9.8015e-01  6.7698e-02 -14.4782 < 2.2e-16 ***
## bac08.round1    -6.1579e-02  2.4312e-02  -2.5329  0.011443 *
## bac10.round1    -1.7340e-02  1.7967e-02  -0.9651  0.334715
## perse.round1    -1.8654e-02  1.4628e-02  -1.2753  0.202470
## sbprim.round1    5.3323e-04  2.4549e-02   0.0217  0.982674
## sbsecon.round1   2.0332e-02  2.1426e-02   0.9490  0.342835
## s170plus.round1  2.1953e-01  2.1689e-02  10.1216 < 2.2e-16 ***
## gdl.round1      -2.0924e-02  2.5275e-02  -0.8279  0.407917
## lperc14_24       2.9370e-01  9.2957e-02   3.1595  0.001621 **
## lunem            2.6715e-01  2.4118e-02  11.0768 < 2.2e-16 ***
## lvehicmilespc    1.5353e+00  4.4496e-02  34.5041 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    142.38
## Residual Sum of Squares: 47.242
## R-Squared:      0.66819
## Adj. R-Squared: 0.65851
## F-statistic: 69.0028 on 34 and 1165 DF, p-value: < 2.22e-16
```

```
# check residuals - not white noise, but normal
mod_diag(mod1b)
```





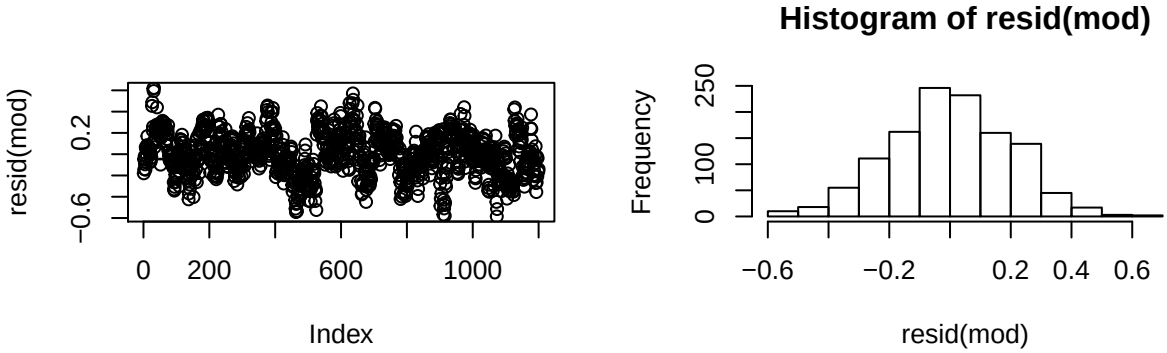

```
# compare pooled and pooled w/ binarized vars models - models look same
stargazer(mod1, mod1b, type = 'text')
```

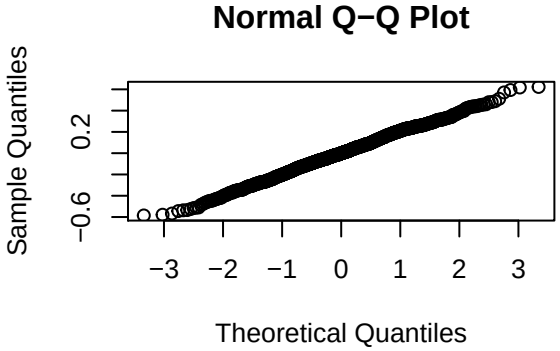
```
##
## ===============================================
##                          Dependent variable:
##                      ---------------------------
##                              ltotfatrte
##                         (1)            (2)
## -----------------------------------------------
## d81                   -0.092**       -0.093**
##                       (0.041)        (0.041)
##
## d82                   -0.296***      -0.297***
##                       (0.042)        (0.042)
##
## d83                   -0.350***      -0.354***
##                       (0.043)        (0.042)
##
## d84                   -0.300***      -0.302***
##                       (0.043)        (0.043)
##
## d85                   -0.338***      -0.339***
##                       (0.044)        (0.044)
##
## d86                   -0.314***      -0.315***
##                       (0.046)        (0.046)
##
## d87                   -0.350***      -0.351***
```

```
##                                (0.048)        (0.048)
##
## d88                           -0.360***      -0.361***
##                                (0.050)        (0.050)
##
## d89                           -0.445***      -0.445***
##                                (0.052)        (0.052)
##
## d90                           -0.504***      -0.504***
##                                (0.053)        (0.053)
##
## d91                           -0.617***      -0.619***
##                                (0.054)        (0.054)
##
## d92                           -0.723***      -0.724***
##                                (0.055)        (0.055)
##
## d93                           -0.714***      -0.716***
##                                (0.056)        (0.056)
##
## d94                           -0.700***      -0.703***
##                                (0.057)        (0.057)
##
## d95                           -0.680***      -0.678***
##                                (0.059)        (0.059)
##
## d96                           -0.800***      -0.803***
##                                (0.061)        (0.061)
##
## d97                           -0.820***      -0.812***
##                                (0.062)        (0.062)
##
## d98                           -0.863***      -0.860***
##                                (0.063)        (0.063)
##
## d99                           -0.864***      -0.862***
##                                (0.064)        (0.064)
##
## d00                           -0.877***      -0.875***
##                                (0.065)        (0.065)
##
## d01                           -0.928***      -0.931***
##                                (0.066)        (0.065)
##
## d02                           -0.972***      -0.975***
##                                (0.067)        (0.066)
##
## d03                           -0.992***      -0.999***
##                                (0.067)        (0.066)
##
## d04                           -0.978***      -0.980***
##                                (0.069)        (0.068)
##
## bac08                          -0.063**
```

```
##                                (0.027)
##
## bac10                          -0.019
##                                (0.020)
##
## perse                          -0.020
##                                (0.015)
##
## sbprim                          0.0001
##                                (0.025)
##
## sbsecon                         0.020
##                                (0.021)
##
## sl70plus                       0.230***
##                                (0.022)
##
## gdl                            -0.027
##                                (0.026)
##
## bac08.round1                                      -0.062**
##                                                    (0.024)
##
## bac10.round1                                       -0.017
##                                                    (0.018)
##
## perse.round1                                       -0.019
##                                                    (0.015)
##
## sbprim.round1                                       0.001
##                                                    (0.025)
##
## sbsecon.round1                                      0.020
##                                                    (0.021)
##
## s170plus.round1                                   0.220***
##                                                    (0.022)
##
## gdl.round1                                         -0.021
##                                                    (0.025)
##
## lperc14_24                     0.288***           0.294***
##                                (0.093)             (0.093)
##
## lunem                          0.264***           0.267***
##                                (0.024)             (0.024)
##
## lvehicmilespc                  1.532***           1.535***
##                                (0.044)             (0.044)
##
## Constant                      -11.717***         -11.771***
##                                (0.445)             (0.444)
##
## ------------------------------------------------------------
```

```
## Observations                    1,200          1,200
## R2                              0.670          0.668
## Adjusted R2                     0.660          0.659
## F Statistic (df = 34; 1165)   69.506***      69.003***
## ==========================================================
## Note:                         *p<0.1; **p<0.05; ***p<0.01
```





How are the variables *bac8* and *bac10* defined? Interpret the coefficients on *bac8* and *bac10*.

- The variables bac08 and bac10 are defined as fractional years during which the BAC limits were 0.08 and 0.10, respectively. In some state-years, the BAC limits changed in the middle of the year. The coefficients for bac08 and bac10 can be interpreted as the percentage change in the fatality rate for each fraction of a year that the BAC laws are in place. In this case, if BAC limits are 0.08 for an entire year, then the model implies that fatality rates will drop by 6.3%. If the BAC limits were 0.10 for an entire year, the fatality rate would drop by 1.9%. However, only bac08 is statistically insignificant (at $\alpha = 0.05$, p-value = 0.011 for bac08 and 0.33 for bac10), so we're not confident in this conclusion for bac10.
- The pooled OLS models estimates a negative effect on both blood alcohol limits of .08 and .10 though the effects are marginal and only the .08 effect reaches a statistical significance level of .05.

Do *per se laws* have a negative effect on the fatality rate? What about having a primary seat belt law? (Note that if a law was enacted sometime within a year the fraction of the year is recorded in place of the zero-one indicator.)

- Per se laws are estimated to have a negative effect on the fatality rate but this estimate does not reach statistical significance in the pooled model and thus we cannot conclude if it is distinguishable from no effect. The estimate of the effect on a primary seat belt law is even more inconclusive with our standard errors on that estimate far greater that the estimate coefficient so we cannot say with any confidence based on this output whether there is an effect.

- Per se laws decrease the fatality rate by about 1.9%. The coefficient for perse is not statistically significant (at $\alpha = 0.05$, p-value = 0.11). Primary seat belt laws appear to have little to no effect on fatality rate as its coefficient is approximately 0. The coefficient for sbprim is not statistically significant (at $alpha = 0.05$, p-value = 0.98).

4. Reestimate the model from *Exercise 3* using a fixed effects (at the state level) model. How do the coefficients on *bac08, bac10, perse, and sbprim* compare with the pooled OLS estimates? Which set of estimates do you think is more reliable?

- The estimate bac08 and bac10 from fixed effect model are less negative than in the pooled OLS and are both insignificant (at $\alpha = 0.05$). The coefficient estimate for perse is -0.059, less than in the pooled OLS, and is highly significant (at $\alpha = 0.001$). The estimate for sbprim is -0.04, again less than in the pooled OLS and highly signifcant (at $\alpha = 0.001$).
- The fixed effect estimates are probably more reliable because it assumes that there is independence between our observations and thus does not account for any of the unobserved heterogeneity whereas the fixed effects model attempts to eliminate this by estimated an average effect per state. The additional assumption of the fixed effect model is that this unobserved heterogeneity is correlate with the other explanatory variables which seems reasonable given that states and the laws they pass would most likely be correlated across observations.
- The fixed effects model also has a higher R-squared (0.71 vs. 0.66), so more variance is explained with the FE estimators. Also, the estimates for bac10, perse, and sbprim are larger in magnitude and/or highly significant, whereas they weren't before, indicating that state unobserved effects may have been correlated with those variables. One explanation is that policies might have a greater effect on safety when there are unsafe driving conditions. Thus, states with rainy climates or hilly, windy roads could experience a greater reduction in fatality rate than states with dry climates and flat roads when laws for wearing seat belts and revoking drivers licenses.

What assumptions are needed in each of these models? Are these assumptions reasonable in the current context?

- The pooled OLS model assumes that the response variable is normally distributed and errors from the fitted model are uncorrelated with the explanatory variables and homoskedastic. Based on the residual diagonostics, this assumption is valid.
- The fixed effects model assumes that the state fixed effects are time-independent and thus can be differenced away. For fixed effects like climate and terrain, this seems reasonable. The model allows for fixed effects to be correlated with the explanatory variables. However, the errors must still be homoskedastic and serially uncorrelated over time.

```
# fit model - fixed effects at state level
mod2 = plm(form1b, data=df.plm, model='within')
summary(mod2)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = form1b, data = df.plm, model = "within")
##
## Balanced Panel: n=48, T=25, N=1200
##
## Residuals :
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -0.3800463 -0.0515324  0.0031357  0.0546621  0.2894414
##
```

```
## Coefficients :
##                   Estimate Std. Error  t-value  Pr(>|t|)
## d81             -0.0637715  0.0180182  -3.5393 0.0004177 ***
## d82             -0.1385593  0.0188839  -7.3374 4.176e-13 ***
## d83             -0.1741675  0.0193635  -8.9946 < 2.2e-16 ***
## d84             -0.2086915  0.0199521 -10.4596 < 2.2e-16 ***
## d85             -0.2320587  0.0208486 -11.1306 < 2.2e-16 ***
## d86             -0.1931853  0.0223796  -8.6322 < 2.2e-16 ***
## d87             -0.2378615  0.0242865  -9.7940 < 2.2e-16 ***
## d88             -0.2681307  0.0266013 -10.0796 < 2.2e-16 ***
## d89             -0.3397672  0.0284165 -11.9567 < 2.2e-16 ***
## d90             -0.3497444  0.0295684 -11.8283 < 2.2e-16 ***
## d91             -0.3851628  0.0302698 -12.7243 < 2.2e-16 ***
## d92             -0.4454106  0.0313835 -14.1925 < 2.2e-16 ***
## d93             -0.4619480  0.0320192 -14.4272 < 2.2e-16 ***
## d94             -0.4966624  0.0329158 -15.0889 < 2.2e-16 ***
## d95             -0.4926601  0.0341118 -14.4425 < 2.2e-16 ***
## d96             -0.5437094  0.0361775 -15.0289 < 2.2e-16 ***
## d97             -0.5654371  0.0371673 -15.2133 < 2.2e-16 ***
## d98             -0.6198019  0.0381587 -16.2427 < 2.2e-16 ***
## d99             -0.6397549  0.0386089 -16.5701 < 2.2e-16 ***
## d00             -0.6729619  0.0391508 -17.1890 < 2.2e-16 ***
## d01             -0.6445962  0.0392164 -16.4369 < 2.2e-16 ***
## d02             -0.6070783  0.0394582 -15.3854 < 2.2e-16 ***
## d03             -0.6103003  0.0396260 -15.4015 < 2.2e-16 ***
## d04             -0.6468036  0.0407634 -15.8673 < 2.2e-16 ***
## bac08.round1    -0.0203730  0.0143928  -1.4155 0.1572002
## bac10.round1    -0.0165389  0.0098247  -1.6834 0.0925762 .
## perse.round1    -0.0531433  0.0097383  -5.4572 5.956e-08 ***
## sbprim.round1   -0.0408466  0.0149669  -2.7291 0.0064499 **
## sbsecon.round1   0.0055382  0.0109861   0.5041 0.6142839
## s170plus.round1  0.0688343  0.0113895   6.0437 2.050e-09 ***
## gdl.round1      -0.0148566  0.0121973  -1.2180 0.2234746
## lperc14_24       0.3407440  0.0627205   5.4327 6.807e-08 ***
## lunem           -0.1909612  0.0171528 -11.1329 < 2.2e-16 ***
## lvehicmilespc    0.6684077  0.0507619  13.1675 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    31.924
## Residual Sum of Squares: 8.655
## R-Squared:      0.72889
## Adj. R-Squared: 0.70925
## F-statistic: 88.4046 on 34 and 1118 DF, p-value: < 2.22e-16
```

```
# check residuals
mod_diag(mod2)

# compare pooled and FE models
stargazer(mod1b, mod2, type = 'text')
```

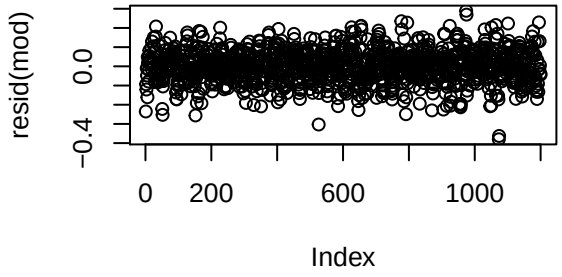
```
##
## ===============================================================
##                                 Dependent variable:
##                 ----------------------------------------------
```

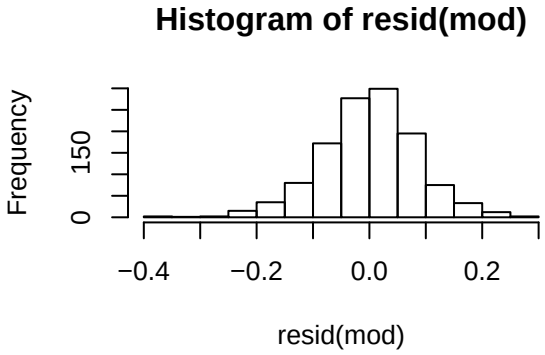
```
##                                    ltotfatrte
##                            (1)                     (2)
## ------------------------------------------------------------------
## d81                     -0.093**                -0.064***
##                          (0.041)                 (0.018)
##
## d82                     -0.297***               -0.139***
##                          (0.042)                 (0.019)
##
## d83                     -0.354***               -0.174***
##                          (0.042)                 (0.019)
##
## d84                     -0.302***               -0.209***
##                          (0.043)                 (0.020)
##
## d85                     -0.339***               -0.232***
##                          (0.044)                 (0.021)
##
## d86                     -0.315***               -0.193***
##                          (0.046)                 (0.022)
##
## d87                     -0.351***               -0.238***
##                          (0.048)                 (0.024)
##
## d88                     -0.361***               -0.268***
##                          (0.050)                 (0.027)
##
## d89                     -0.445***               -0.340***
##                          (0.052)                 (0.028)
##
## d90                     -0.504***               -0.350***
##                          (0.053)                 (0.030)
##
## d91                     -0.619***               -0.385***
##                          (0.054)                 (0.030)
##
## d92                     -0.724***               -0.445***
##                          (0.055)                 (0.031)
##
## d93                     -0.716***               -0.462***
##                          (0.056)                 (0.032)
##
## d94                     -0.703***               -0.497***
##                          (0.057)                 (0.033)
##
## d95                     -0.678***               -0.493***
##                          (0.059)                 (0.034)
##
## d96                     -0.803***               -0.544***
##                          (0.061)                 (0.036)
##
## d97                     -0.812***               -0.565***
##                          (0.062)                 (0.037)
##
```

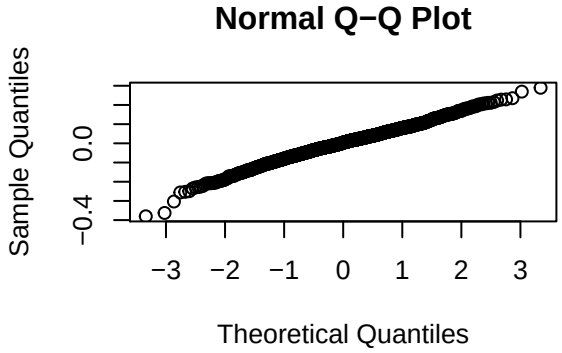
```
## d98                    -0.860***              -0.620***
##                         (0.063)                (0.038)
##
## d99                    -0.862***              -0.640***
##                         (0.064)                (0.039)
##
## d00                    -0.875***              -0.673***
##                         (0.065)                (0.039)
##
## d01                    -0.931***              -0.645***
##                         (0.065)                (0.039)
##
## d02                    -0.975***              -0.607***
##                         (0.066)                (0.039)
##
## d03                    -0.999***              -0.610***
##                         (0.066)                (0.040)
##
## d04                    -0.980***              -0.647***
##                         (0.068)                (0.041)
##
## bac08.round1           -0.062**                -0.020
##                         (0.024)                (0.014)
##
## bac10.round1            -0.017                 -0.017*
##                         (0.018)                (0.010)
##
## perse.round1            -0.019                 -0.053***
##                         (0.015)                (0.010)
##
## sbprim.round1            0.001                 -0.041***
##                         (0.025)                (0.015)
##
## sbsecon.round1           0.020                  0.006
##                         (0.021)                (0.011)
##
## s170plus.round1        0.220***               0.069***
##                         (0.022)                (0.011)
##
## gdl.round1              -0.021                 -0.015
##                         (0.025)                (0.012)
##
## lperc14_24             0.294***               0.341***
##                         (0.093)                (0.063)
##
## lunem                  0.267***               -0.191***
##                         (0.024)                (0.017)
##
## lvehicmilespc          1.535***               0.668***
##                         (0.044)                (0.051)
##
## Constant              -11.771***
##                         (0.444)
##
```

```
## ------------------------------------------------------------------
## Observations              1,200                     1,200
## R2                        0.668                     0.729
## Adjusted R2               0.659                     0.709
## F Statistic     69.003*** (df = 34; 1165) 88.405*** (df = 34; 1118)
## ==================================================================
## Note:                                   *p<0.1; **p<0.05; ***p<0.01
```

```
# # create dummy variables for each state
# states = unique(data$state)
# df.states = setNames(lapply(states, function(x) as.numeric(data$state==x)), paste('s', states, sep =
# df.states = pdata.frame(cbind(data[c('state','year')], df.states))
#
# # add dummay variables to panel df
# df.plm = pdata.frame(cbind(df.plm, df.states), index=c('state','year'))
# # head(df.plm); tail(df.plm)
```







5. Would you prefer to use a random effects model instead of the fixed effects model you build in *Exercise 4*? Why? Why not?

- A random effects model assumes that the state fixed effects are not correlated with the explanatory variables. This assumption is probably not valid though. The fixed effect model explained more variance in the data and resulted in highly significant estimates, implying some correlation between the fixed effects and explanatory variables. Thus, we would not use the random effects model.

6. Suppose that *vehicmilespc*, the number of miles driven per capita, increases by 1,000. Using the FE estimates, what is the estimated effect on totfatrte? Be sure to interpret the estimate as if explaining to a layperson.

- If the miles driven per capita were to increase by 1000, then the fatality rate would only decrease by 0.93%.

```
# change in fat. rate after taking inverse of log-log transform
exp(.668*log(1000))-100
```

```
## [1] 0.9252886
```

7. If there is serial correlation or heteroskedasticity in the idiosyncratic errors of the model, what would be the consequences on the coefficient estimates and their standard errors?

- In the presence of serial correlation or heteroskedasticity our coefficient estimators remain unbiased assuming our endogeneity assumption holds but their standard errors are not estimated properly. If a positive serial correlation exists between errors then we are underestimating our standard errors and thus rendering most statisticals tests invalid. While unlikely it is possible that there is negative serial correlation which would make estimating the effect on the standard errors more difficult but most certainly does not lead to efficient estimates.