Lab 4 - Panel Data Modelling: Fixed and Random Effects

Nishant Velagapudi, Bas Hendri, Brandon Cummings

April 22, 2018

1. Exploratory Data Analysis

We first load the data and separate out variables into potential categories. We know that our response variable is total fatality rate - representing the number of accident fatalities per 100,000 population. We have some suggested explanatory variables for expanding our model in subsequent sessions and so group these together. Finally, we explicitly identify and separate the boolean year flags, potential index columns, and legislature booleans that were excluded from the suggested set of explanatory variables.

```
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(col)
        sapply(col$counts, as.numeric)))) %>%
        inner_join(desc %>% mutate_if(is.factor, as.character), y=.) %>%
        select(-matches('\\.\\d\\d')) %>%
        filter(variable %in% setdiff(a.vars, d.vars))

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

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

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)
```

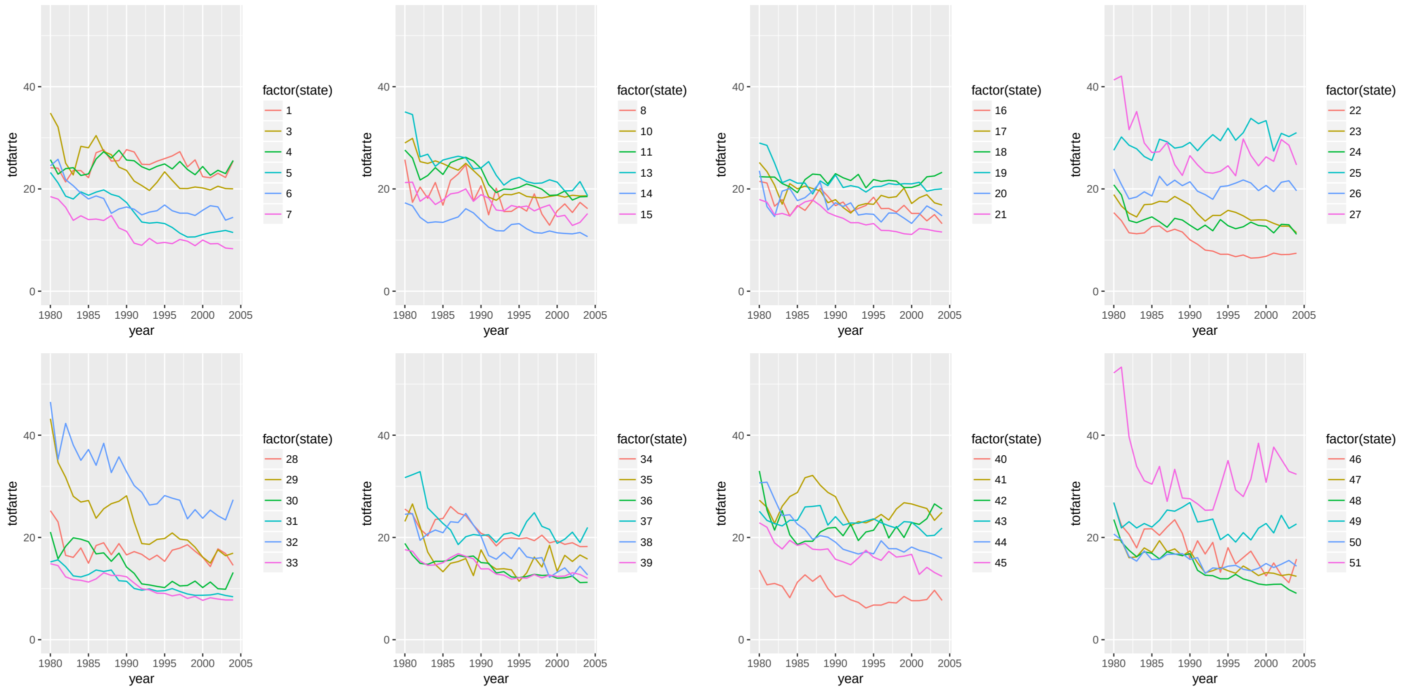
```
##         n   missing  distinct      Info       Sum      Mean       Gmd
##    "1200"       "0"       "2"   "0.115"      "48"    "0.04" "0.07686"
```

```
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(colour=factor(state)))
        + ylim(0, ymax)) + theme(legend.position = 'bottom')  }
```

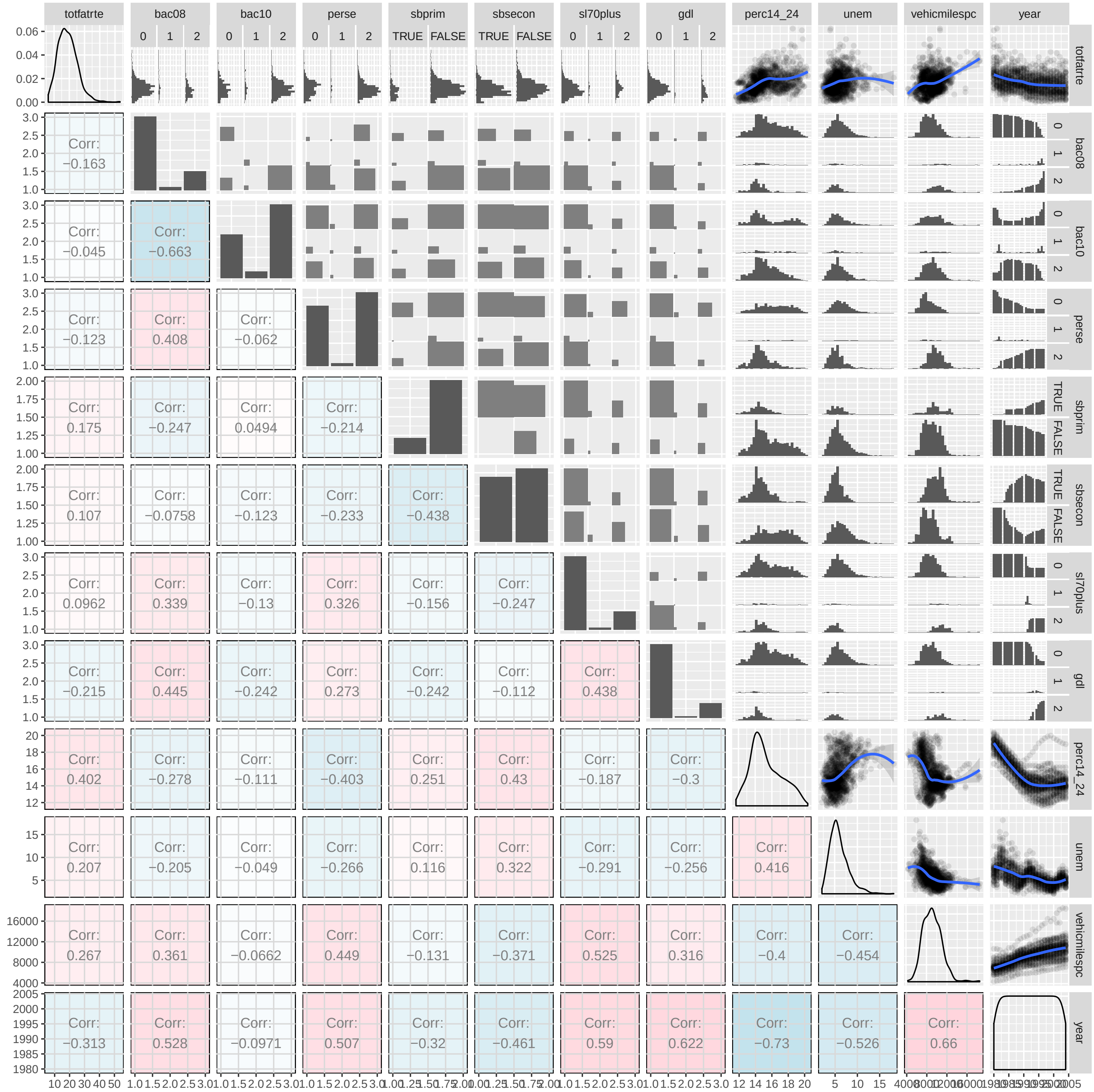


There are relatively few indicator values that are neither 0 nor 1, but these few could be problematic. If we treat the indicator as a factor, these values will introduce a number of distinct levels with small samples. If we treat this indicator as linear, there is an implication of linearity between 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.

```r
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))
```

With these transformations, we plot a matrix of plots and correlation coefficients. Plot type was determined by the variables involved. Loess curves are added in relevant cases.

```r
points_with_loess <- function(data, mapping, method="loess", ...){
  return(ggplot(data=data, mapping=mapping) +
      geom_point(alpha=0.1, ...) +
      geom_smooth(method=method, mapping=mapping, ...)  +
        coord_cartesian(xlim=range(data[,as.character(mapping$x)]),
                        ylim=range(data[,as.character(mapping$y)])))
}
coerced_cor <- function(data, mapping, ...){
  coerced.data <- data %>% mutate_if(is.factor, as.numeric)
  cor.value <- cor(coerced.data[,as.character(mapping$x)], coerced.data[,as.character(mapping$y)])
  ggally_cor(coerced.data, mapping) +
    theme(panel.background=element_rect(fill=do.call(rgb,
          list(colorRamp(c("lightblue", "white", "pink"))((cor.value+1)/2)/255))))
}
r.vars %>% union(e.vars) %>% union(c('year')) %>% ggpairs(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)
    ), columns=.,
  lower=list(continuous=coerced_cor, combo=coerced_cor, discrete=coerced_cor),
  upper=list(continuous=points_with_loess, combo='facethist', discrete='ratio'))
```



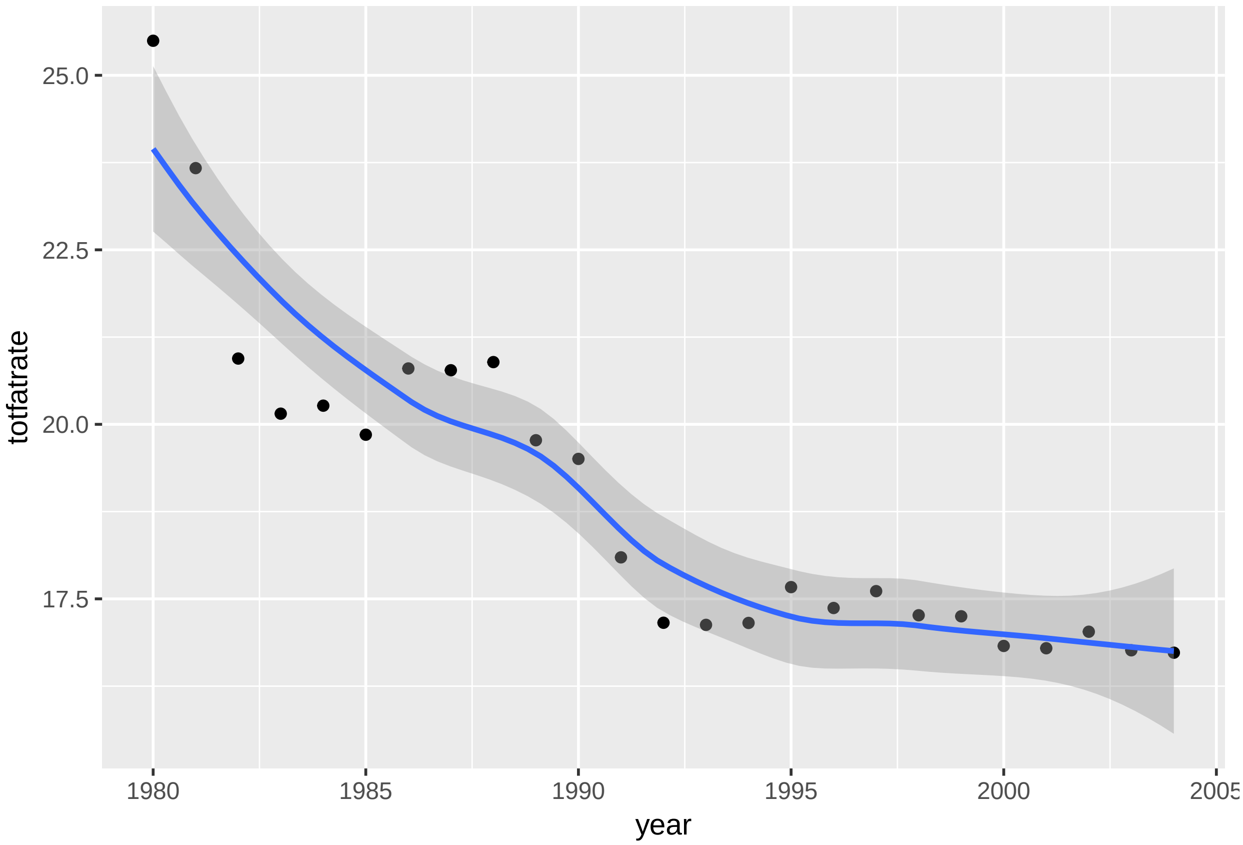
Of particular interest are the trends in bac08 and bac10. We can see that over time, the number of states with an allowable bac of 10 falls, while the number of states with an allowable BAC of .08 rises. We can also see that the number of sl70plus and graduated driverse license law states (gdl) have steep inclines at around the year 1995.

After transforming the indicator variables as described, we would argue that no other transformations are needed. With a sample size of 1200 and near-normal distributions of numeric variables, we expect that the raw data will behave acceptably.

2. Average 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.

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



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: 0x000000002c891e60>
```

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

```
##
## Call:
## lm(formula = simple.formula, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9302  -4.3468  -0.7305   3.7488  29.6498
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.72896    0.86712  19.293  < 2e-16 ***
## d80          8.76563    1.22629   7.148 1.54e-12 ***
## d81          6.94125    1.22629   5.660 1.90e-08 ***
## d82          4.21354    1.22629   3.436 0.000611 ***
## d83          3.42396    1.22629   2.792 0.005321 **
## d84          3.53854    1.22629   2.886 0.003979 **
## d85          3.12250    1.22629   2.546 0.011014 *
## d86          4.07146    1.22629   3.320 0.000927 ***
## d87          4.04583    1.22629   3.299 0.000999 ***
## d88          4.16271    1.22629   3.395 0.000710 ***
## d89          3.04333    1.22629   2.482 0.013213 *
## d90          2.77625    1.22629   2.264 0.023759 *
## d91          1.36583    1.22629   1.114 0.265596
## d92          0.42896    1.22629   0.350 0.726550
## d93          0.39875    1.22629   0.325 0.745112
## d94          0.42625    1.22629   0.348 0.728208
## d95          0.93958    1.22629   0.766 0.443712
## d96          0.64042    1.22629   0.522 0.601603
## d97          0.88167    1.22629   0.719 0.472302
## d98          0.53646    1.22629   0.437 0.661855
## d99          0.52146    1.22629   0.425 0.670745
## d00          0.09667    1.22629   0.079 0.937183
## d01          0.06375    1.22629   0.052 0.958549
## d02          0.30062    1.22629   0.245 0.806383
## d03          0.03458    1.22629   0.028 0.977506
## d04               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.008 on 1175 degrees of freedom
## Multiple R-squared:  0.1276, Adjusted R-squared:  0.1098
## F-statistic: 7.164 on 24 and 1175 DF,  p-value: < 2.2e-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.

3. Expanded Model

As explained in the EDA section, we have transformed indicator variable values that were between 0 and 1. The rationale for this was that there were few observations at each discrete value: bucketing all discrete values between 0 and 1 gives better estimates of the effect of these laws in transition on total fatality rate.

```
expanded.formula <- update(simple.formula, do.call(paste, c(list(sep='+'), '~ .', e.vars)))
(expanded.lm <- lm(expanded.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)
    ))) %>% summary
```

```
##
## Call:
## lm(formula = expanded.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)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.874  -2.710  -0.260   2.303  21.480
##
## Coefficients: (1 not defined because of singularities)
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.932e+01  2.141e+00  -9.025  < 2e-16 ***
## d80           1.684e+01  1.393e+00  12.090  < 2e-16 ***
## d81           1.466e+01  1.376e+00  10.656  < 2e-16 ***
## d82           1.015e+01  1.377e+00   7.369 3.26e-13 ***
## d83           9.121e+00  1.367e+00   6.673 3.88e-11 ***
## d84           1.089e+01  1.272e+00   8.562  < 2e-16 ***
## d85           1.029e+01  1.234e+00   8.342  < 2e-16 ***
## d86           1.095e+01  1.166e+00   9.391  < 2e-16 ***
## d87           1.044e+01  1.127e+00   9.266  < 2e-16 ***
## d88           1.021e+01  1.091e+00   9.352  < 2e-16 ***
## d89           8.739e+00  1.072e+00   8.155 9.00e-16 ***
## d90           7.847e+00  1.061e+00   7.394 2.73e-13 ***
## d91           5.720e+00  1.055e+00   5.423 7.13e-08 ***
## d92           3.921e+00  1.052e+00   3.727 0.000203 ***
## d93           4.054e+00  1.042e+00   3.891 0.000106 ***
## d94           4.417e+00  1.027e+00   4.302 1.84e-05 ***
## d95           4.498e+00  1.023e+00   4.397 1.20e-05 ***
## d96           2.600e+00  1.052e+00   2.472 0.013570 *
## d97           2.443e+00  9.684e-01   2.523 0.011774 *
## d98           1.780e+00  9.390e-01   1.896 0.058264 .
## d99           1.750e+00  9.026e-01   1.939 0.052776 .
## d00           1.407e+00  8.874e-01   1.585 0.113236
## d01           5.576e-01  8.560e-01   0.651 0.514902
## d02           2.771e-02  8.401e-01   0.033 0.973692
## d03          -3.652e-01  8.435e-01  -0.433 0.665173
## d04                  NA         NA      NA       NA
## bac081       -1.566e+00  1.150e+00  -1.362 0.173341
## bac082       -2.366e+00  5.405e-01  -4.377 1.31e-05 ***
## bac101        2.475e-01  9.448e-01   0.262 0.793394
## bac102       -1.368e+00  3.994e-01  -3.426 0.000635 ***
## perse1       -3.041e-01  8.353e-01  -0.364 0.715886
## perse2       -6.401e-01  2.988e-01  -2.142 0.032401 *
## sbprimFALSE   9.661e-02  4.911e-01   0.197 0.844080
## sbseconFALSE -6.625e-02  4.296e-01  -0.154 0.877464
## sl70plus1     2.952e+00  9.185e-01   3.214 0.001346 **
## sl70plus2     3.337e+00  4.487e-01   7.437 2.00e-13 ***
## gdl1         -7.529e-01  9.739e-01  -0.773 0.439641
## gdl2         -4.117e-01  5.301e-01  -0.776 0.437615
## perc14_24     1.315e-01  1.229e-01   1.070 0.284917
## unem          7.577e-01  7.804e-02   9.710  < 2e-16 ***
## vehicmilespc  2.912e-03  9.527e-05  30.568  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.048 on 1160 degrees of freedom
## Multiple R-squared:  0.609,  Adjusted R-squared:  0.5959
## F-statistic: 46.34 on 39 and 1160 DF,  p-value: < 2.2e-16
```

We observe that the effect of the dummy year variables remains similar to what was observed with the model only using these indicators. Once again, in each of our ternary factors, baseline is the 0 value, whereas 1 indicates a transitive value and 2 indicates that the legislature was in place for the full year. We can see that bac08, bac10, and perse have negative coefficients (thus reducing total fatality rate) and are statistically significant when in place for a full year. We observe that the sl70plus laws actually increase total fatality rate both in transition and when in place for a full year. We observe that unemployment and vehicle miles per capita also have significant effects. Interestingly, neither the primary nor secondary seatbelt laws appear to have a significant effect on total fatality rate: the baseline effect being that the laws are inplace.

4: Fixed 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: 0x0000000028aa9dc8>
```

```
(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.ternary.factor(bac08),
    bac10=as.ternary.factor(bac10),
    perse=as.ternary.factor(perse),
    gdl=as.ternary.factor(gdl),
    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.
## -7.479236 -1.183526 -0.073681  1.118514 14.385238
##
## Coefficients:
##                 Estimate  Std. Error  t-value  Pr(>|t|)
## bac081       -0.41088752  0.60583937  -0.6782 0.4977753
## bac082       -1.89728800  0.38399310  -4.9409 8.940e-07 ***
## bac101       -0.76521734  0.49546657  -1.5444 0.1227606
## bac102       -1.45176041  0.26865561  -5.4038 7.944e-08 ***
## perse1       -1.12894730  0.46581393  -2.4236 0.0155227 *
## perse2       -1.55370602  0.24826216  -6.2583 5.503e-10 ***
## sbprimFALSE   1.80989993  0.34493434   5.2471 1.842e-07 ***
## sbseconFALSE  0.86212401  0.24809475   3.4750 0.0005301 ***
## sl70plus1    -0.70047158  0.44340189  -1.5798 0.1144384
## sl70plus2    -1.16546049  0.24803833  -4.6987 2.938e-06 ***
## gdl1         -1.27296047  0.51608006  -2.4666 0.0137871 *
## gdl2         -0.61143985  0.23171993  -2.6387 0.0084361 **
## perc14_24     0.94369366  0.07108756  13.2751 < 2.2e-16 ***
## unem         -0.59167682  0.05148528 -11.4922 < 2.2e-16 ***
## vehicmilespc  0.00029475  0.00010331   2.8530 0.0044097 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    12134
## Residual Sum of Squares: 5470.7
## R-Squared:      0.54915
## Adj. R-Squared: 0.52456
## F-statistic: 92.3251 on 15 and 1137 DF, p-value: < 2.22e-16
```

5: Random 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.ternary.factor(bac08),
    bac10=as.ternary.factor(bac10),
    perse=as.ternary.factor(perse),
    gdl=as.ternary.factor(gdl),
    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 4.812   2.194  0.37
## individual    8.207   2.865  0.63
## theta: 0.8486
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -6.10068 -1.42722 -0.24709  1.03465 16.66514
##
## Coefficients:
##                 Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)   2.12007898  1.79987376  1.1779 0.2390715
## bac081       -0.51277512  0.63461112 -0.8080 0.4192444
## bac082       -2.16723982  0.39746764 -5.4526 6.037e-08 ***
## bac101       -0.90619575  0.51882732 -1.7466 0.0809621 .
## bac102       -1.55399730  0.27898619 -5.5702 3.148e-08 ***
## perse1       -1.04711946  0.48725545 -2.1490 0.0318354 *
## perse2       -1.45676642  0.25574250 -5.6962 1.544e-08 ***
## sbprimFALSE   1.89390217  0.35630965  5.3153 1.271e-07 ***
## sbseconFALSE  0.98291454  0.25790414  3.8112 0.0001454 ***
## sl70plus1    -0.69556596  0.46442557 -1.4977 0.1344801
## sl70plus2    -1.18555502  0.25805083 -4.5943 4.807e-06 ***
## gdl1         -1.31063241  0.54098035 -2.4227 0.0155550 *
## gdl2         -0.75462127  0.24203151 -3.1179 0.0018656 **
## perc14_24     1.00666816  0.07314819 13.7620 < 2.2e-16 ***
## unem         -0.51268328  0.05327356 -9.6236 < 2.2e-16 ***
## vehicmilespc  0.00054339  0.00010349  5.2505 1.797e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    12970
## Residual Sum of Squares: 6265.3
## R-Squared:      0.51693
## Adj. R-Squared: 0.51081
## F-statistic: 84.4671 on 15 and 1184 DF, p-value: < 2.22e-16
```

6: Fixed Effect With Constant

If we were to increase the average number of miles driven per 100,000 people in every state by 1,000 miles, we would see the following increase in fatalities calculated from our fixed effects model and random effects model:

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

```
## [1] 0.2947534
```

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

```
## [1] 0.543393
```

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.

7: Serial Correlation and Heteroskedasticity

We will use the Breusch-Godfrey/Wooldridge test for serial correlation:

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

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel
##  models
##
## data:  effects.formula
## chisq = 395.31, df = 25, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

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

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel
##  models
##
## data:  effects.formula
## chisq = 440.95, df = 25, p-value < 2.2e-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:

```
bptest(fe.plm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fe.plm
## BP = 95.802, df = 15, p-value = 8.104e-14
```

```
bptest(re.plm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  re.plm
## BP = 95.802, df = 15, p-value = 8.104e-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       -0.41088752  0.62109325 -0.6616 0.5083903
## bac082       -1.89728800  0.66081860 -2.8711 0.0041661 **
## bac101       -0.76521734  0.44945639 -1.7025 0.0889276 .
## bac102       -1.45176041  0.49516208 -2.9319 0.0034365 **
## perse1       -1.12894730  0.45803570 -2.4648 0.0138576 *
## perse2       -1.55370602  0.45042038 -3.4495 0.0005823 ***
## sbprimFALSE   1.80989993  0.74555097  2.4276 0.0153535 *
## sbseconFALSE  0.86212401  0.47396989  1.8189 0.0691831 .
## sl70plus1    -0.70047158  0.38947903 -1.7985 0.0723656 .
## sl70plus2    -1.16546049  0.55390642 -2.1041 0.0355911 *
## gdl1         -1.27296047  0.55763105 -2.2828 0.0226261 *
## gdl2         -0.61143985  0.33741460 -1.8121 0.0702295 .
## perc14_24     0.94369366  0.17063636  5.5304 3.961e-08 ***
## unem         -0.59167682  0.07859314 -7.5284 1.043e-13 ***
## vehicmilespc  0.00029475  0.00027079  1.0885 0.2766082
## ---
## 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)   2.12007898  4.15447475  0.5103  0.609928
## bac081       -0.51277512  0.61993243 -0.8271  0.408321
## bac082       -2.16723982  0.68235034 -3.1761  0.001531 **
## bac101       -0.90619575  0.47373770 -1.9129  0.056007 .
## bac102       -1.55399730  0.51495100 -3.0178  0.002601 **
## perse1       -1.04711946  0.44601705 -2.3477  0.019053 *
## perse2       -1.45676642  0.42991926 -3.3885  0.000726 ***
## sbprimFALSE   1.89390217  0.72908292  2.5976  0.009503 **
## sbseconFALSE  0.98291454  0.47857303  2.0538  0.040210 *
## sl70plus1    -0.69556596  0.37930607 -1.8338  0.066937 .
## sl70plus2    -1.18555502  0.52992183 -2.2372  0.025457 *
## gdl1         -1.31063241  0.55510518 -2.3611  0.018384 *
## gdl2         -0.75462127  0.33439007 -2.2567  0.024208 *
## perc14_24     1.00666816  0.16877934  5.9644 3.238e-09 ***
## unem         -0.51268328  0.07965307 -6.4365 1.773e-10 ***
## vehicmilespc  0.00054339  0.00024939  2.1789  0.029534 *
## ---
## 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.