W271 Lab 3

Adam Sayre, Luke Verdi, & Erin Werner

December 12th, 2020

Contents


```
library(ggplot2)
library(gplots)
library(dplyr)
library(gridExtra)
library(plm)
library(Hmisc)
library(maps)
library(mapproj)
```

U.S. traffic fatalities: 1980-2004

In this lab, we are asked to answer the question **"Do changes in traffic laws affect traffic fatalities?"** To do so, we 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 economic and demographic variables are also included. The description of the each of the variables in the data set comes with the data.

Question 1

Load the data. Provide a description of the basic structure of the data set, as we have done throughout the semester. Conduct a very thorough EDA, which should include both graphical and tabular techniques, on the data set, including both the dependent variable `totfatrte` and the potential explanatory variables. You need to write a detailed narrative of your observations of your EDA.

First, we can load the data. This data is structured as a long panel data 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.

```
load(paste0(here::here(),"/labs/lab_3/driving.RData"))
driving <- data
```

We can produce a table to better understand the dimensions of our panel data. In this context, our time variant is `year` and our cross-sectional variant is `state`.

```
table(driving$state)
```

```
##
##  1  3  4  5  6  7  8 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
## 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25
## 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
## 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25
```

From the table, we can see that there are indeed 48 states with values recorded over 25 years.

Overall, the data set includes many variables to explore. By reading about them in the `desc` data frame loaded along with the data, we see that many of the variables are indicator variables for the different years (25 variables for years 1980 to 2004). There are many other indicator variables as well, such as for different speed limits (`sl55`, `sl65`, `sl70`, `sl75`, `slnone` and `sl70plus`), for seatbelt laws (`seatbelt`, `sbprim` and `sbsecon`), or for the presence of other various laws (`zerotol`, `gdl`, `bac10`, `bac08`, `perse`). The data set also includes continuous variables, like the minimum drinking age (`minage`), the vehicle miles traveled in billions (`vehicmiles`) or the vehicle miles per car (`vehicmilespc`), the state unemployment rate (`unem`), and the percentage of population aged 14-24 (`perc14_24`). Lastly, the data set includes our response variable of interest, the total fatality rate (`totfatrte`), but also the night fatality rate (`nghtfatrte`) and weekend fatality rate (`wkndfatrte`). These variables are also represented in an absolute form (`totfat`, `nghtfat`, `wkndfat`) and in a miles driven normalized form (`totfatpvm`, `nghtfatpvm`, `wkndfatpvm`).

As there are many variables in the data set to explore, we will focus on the ones that are deemed most relevant to our analysis. This includes the independent variable `totfatrte` and the dependent variables `bac08`, `bac10`, `perse`, `sbprim`, `sbsecon`, `sl70plus`, `gdl`, `perc14_24`, `unem`, and `vehicmilespc`. We've chosen these as they intuitively seem to bring down the total fatalities brought about by driving. Noticeably, we've omitted many variables that are in the data set. But, we have done this only in situations where the variables are also a likely dependent variable in this context (most obviously `totfat`, but also something like `wkndfat` or even `wkndfatpvm`). We've also omitted variables that are already somewhat covered by the other included variables. An example of this would be how we think having a very high speed limit or no speed limit might impact driving fatalities (i.e. `sl70plus`), and so we have chosen to omit the granularity about whether the highest speed limit is 55, 65 or 70. Therefore, our EDA will focus on these particular variables.

Now, we can take an overall summary look at some of the binary variables of interest.

```
summary(driving[,c(13,12,29,30,28)], digits = 2)
```

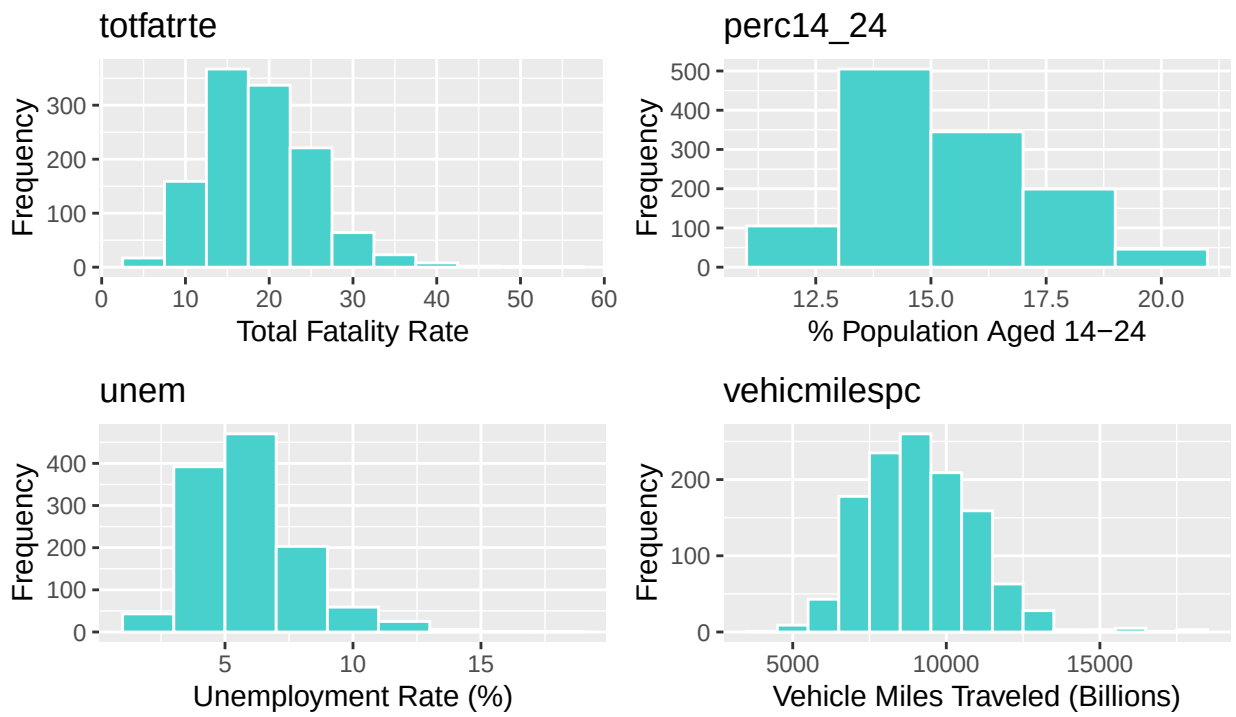
```
##      bac08          bac10          sbprim        sbsecon        sl70plus
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
```

```
##  1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.00
##  Median :0.00   Median :1.00   Median :0.00   Median :0.00   Median :0.00
##  Mean   :0.21   Mean   :0.62   Mean   :0.18   Mean   :0.47   Mean   :0.21
##  3rd Qu.:0.00   3rd Qu.:1.00   3rd Qu.:0.00   3rd Qu.:1.00   3rd Qu.:0.00
##  Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
```

Here we see that if a law was enacted sometime within a year the fraction of the year is recorded in place of the zero-one indicator. This will be important to consider in building our model. We can also see that there are many instances where states and years do not have a certain law, as there are means less than 0.5, indicating that there are more 'zero' values compared to 'one' values. We also see periods where there is neither BAC law or neither seat belt law, as the cumulative means are less than 1.

We can then explore the distributions of our non-binary variables.

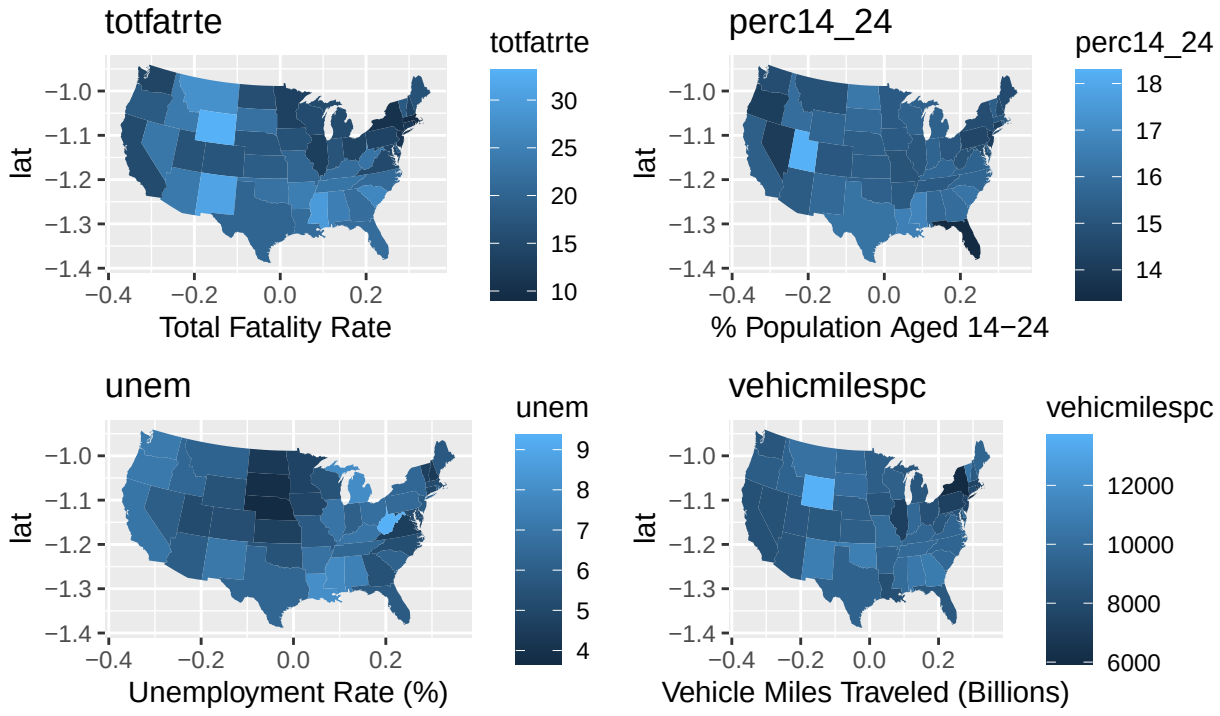
```
p1 <- ggplot(driving,aes(totfatrte)) +
  geom_histogram(binwidth=5,fill="mediumturquoise",col="white",size=0.5)+
  labs(title="totfatrte",x="Total Fatality Rate",y="Frequency")
p2 <- ggplot(driving, aes(perc14_24)) +
  geom_histogram(binwidth = 2, fill = "mediumturquoise",col="white",size = 0.5)+
  labs(title="perc14_24", x = "% Population Aged 14-24",y="Frequency")
p3 <- ggplot(driving, aes(unem)) +
  geom_histogram(binwidth = 2,fill="mediumturquoise",col="white",size=0.5)+
  labs(title="unem", x = "Unemployment Rate (%)",y="Frequency")
p4 <- ggplot(driving, aes(vehicmilespc)) +
  geom_histogram(binwidth=1000,fill="mediumturquoise",col="white",size=0.5) +
  labs(title="vehicmilespc",x="Vehicle Miles Traveled (Billions)",y="Frequency")
egg::ggarrange(p1, p2, p3, p4, nrow = 2)
```



From the histograms, we can see that each distribution is uni-modal with a slight right skew towards lower values. The total fatality rate (`totfatrte`), which is the dependent variable in our model, generally has a value of about 18%. It is important to note that each of these distributions, besides `vehicmilespc`, is scaled as a percent rate.

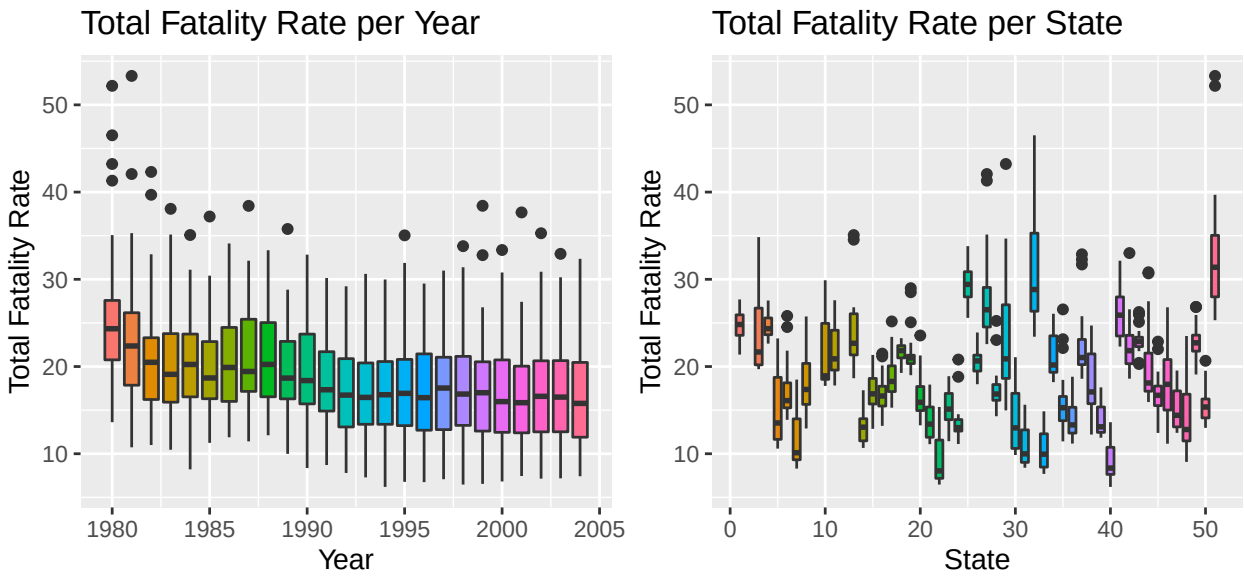
We can also see how these values vary, on average, across each state.

```
states = map_data('state', projection = "albers", parameters = c(39, 45))
sn = unique(states$region[])
sn = c(sn[1], "alaska", sn[2:10], 'hawaii', sn[11:length(sn)])
states = states[states$region != "district of columbia",]
states$state = match(states$region,sn)
t.map = merge(states, driving, by="state")
no_v = !names(t.map)%in%c('year','region','subregion','lat','long','order','group')
traffic.state.agg = aggregate(t.map[, no_v],list(t.map$state),mean)
t.map.agg = merge(states, traffic.state.agg, by="state")
pm1 = qplot(long,lat,data=t.map.agg,geom="polygon",fill=totfatrte,group=group) +
  labs(x = "Total Fatality Rate",title="totfatrte")
pm2 = qplot(long,lat,data=t.map.agg,geom="polygon",fill=perc14_24,group=group) +
  labs(x = "% Population Aged 14-24",title="perc14_24")
pm3 = qplot(long,lat,data=t.map.agg,geom="polygon",fill=unem,group=group) +
  labs(x = "Unemployment Rate (%)",title="unem")
pm4 = qplot(long,lat,data=t.map.agg,geom="polygon",fill=vehicmilespc,group=group) +
  labs(x = "Vehicle Miles Traveled (Billions)",title="vehicmilespc")
grid.arrange(pm1,pm2,pm3,pm4,nrow=2,ncol=2)
```



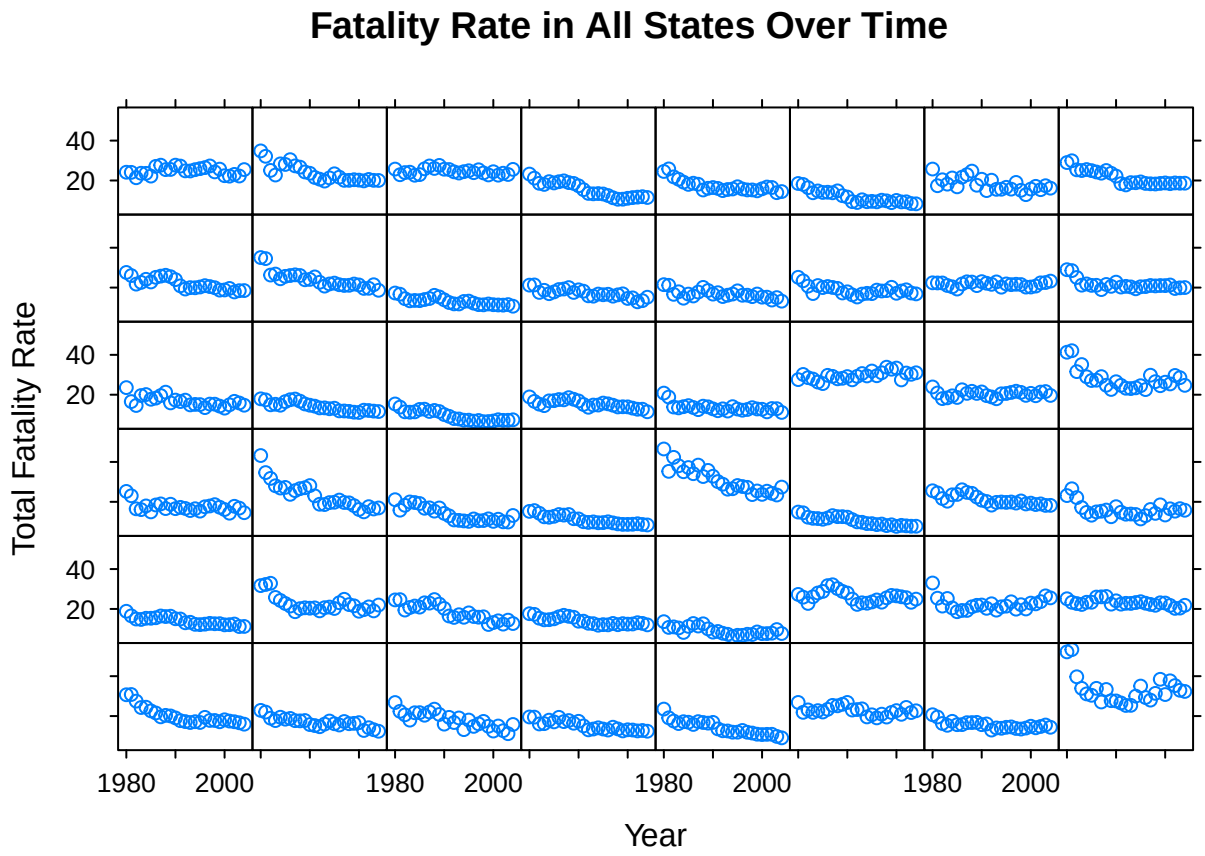
From the maps, we can see that dark blue states indicate lower values whereas lighter blue states have higher values. It appears that Wyoming has similarly high `totfatrte` and `vehicmilespc`. As our dependent variable is `totfatrte`, we can further explore it in relation to both the year and state of our panel data.

```
p1 <- ggplot(driving,aes(year,totfatrte))+theme(legend.position="none") +
  labs(x="Year",y="Total Fatality Rate", title = "Total Fatality Rate per Year") +
  geom_boxplot(varwidth=T, aes(fill = factor(year)))
p2 <- ggplot(driving,aes(state,totfatrte))+theme(legend.position="none") +
  labs(x="State",y="Total Fatality Rate", title = "Total Fatality Rate per State") +
  geom_boxplot(varwidth=T, aes(fill = factor(state)))
egg::ggarrange(p1, p2, nrow = 1)
```



From the box plots, we can see that the total fatality rate has a slightly negative trend as the distribution is decreasing over time. We can also see that the fatality rate varies a lot across each state, as some states have more extreme values than others. We can also get an overall understanding of these three effects together in a Trellis plot. Note that each box is for a separate state and shows the `totfatrte` over time.

```
xyplot(totfatrte ~ year | state, data=driving, as.table=T, strip = F, xlab = 'Year',
       ylab = 'Total Fatality Rate', main = 'Fatality Rate in All States Over Time',
       scales = list(alternating = c(1,0), tick.number = c(2), tck = 0.5))
```



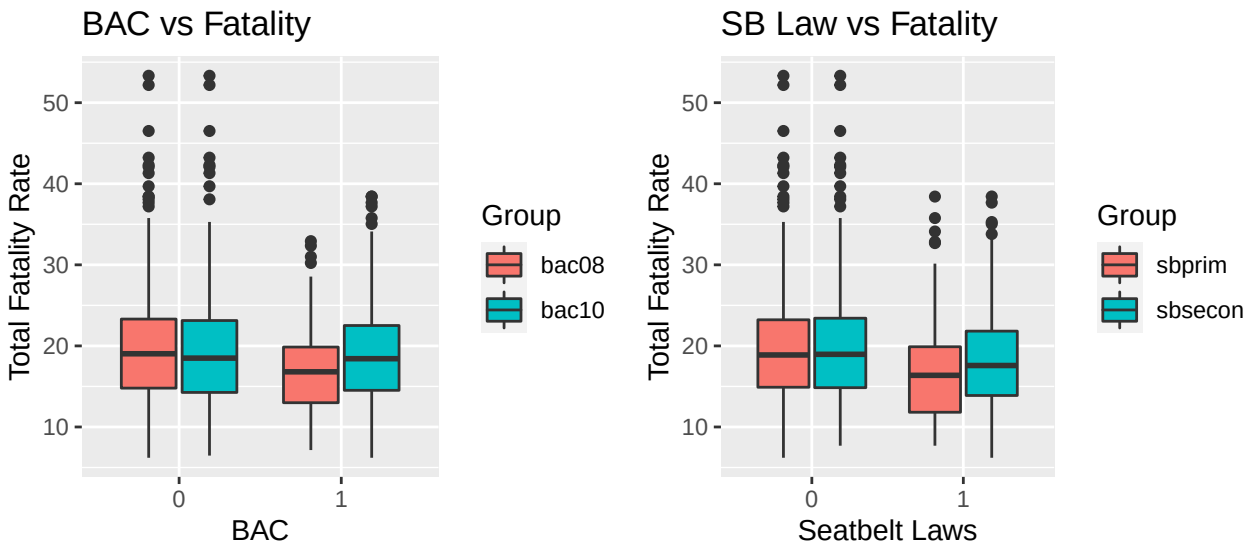
From the plot, we can reaffirm that the fatality rate does generally decline over time and the rate varies from state to state. Though most states seem to show a negative trend, a few exceptions are flat. It doesn't appear that any states showed an increasing trend in driving fatalities.

Now that we have a good understanding of our dependent variable, we can take a closer look at some of our explanatory variables. To start, we can investigate the explanatory variables by their general feature category. As a result, we can compare the `bac10` (blood alcohol limit .10) and the `bac10` (blood alcohol limit .08) variables together as well as the `sbprim` (which = 1 if there is a primary seatbelt law) and `sbsecon` (which = 1 if there is a secondary seatbelt law) laws together.

Note that in order to better generalize our binary variables, we will convert the fraction values to either 'zeros' or 'ones', based on rounding. We believe this rounding makes sense for these variables as it then defaults to whether the law was in place for the majority of the year (rounding up at 6 months) or not. The lone exception for this will be if a state went from one law to another (such as `bac08` to `bac10`) in a month like July. Then, both variables would get rounded to 1 in this model. So, in those edge cases we'll round `bac10` down. However, this is not a very common case and should not have a large impact on our model.

```
driving <- driving %>% mutate(bac10_fac = ifelse(bac10 == 0.5, 0, round(bac10)),
  bac08_fac=round(bac08),perse_fac=round(perse),sbprim_fac=round(sbprim),
  sbsecon_fac=round(sbsecon),sl70plus_fac=round(sl70plus),gdl_fac=round(gdl))
```

```
all_totfatrte <- c(driving$totfatrte,driving$totfatrte)
all_bac <- c(driving$bac08_fac,driving$bac10_fac)
all_group_b <- c(rep("bac08",1200),rep("bac10",1200))
bac_df <- data.frame(BAC=all_bac,totrte=all_totfatrte,Group=all_group_b)
all_sb_law <- c(driving$sbprim_fac,driving$sbsecon_fac)
all_group_s <- c(rep("sbprim",1200),rep("sbsecon",1200))
sb_law_df <- data.frame(SB_Law=all_sb_law,totrte=all_totfatrte,Group=all_group_s)
p1 <- ggplot(bac_df,aes(x=factor(BAC),y=totrte)) + geom_boxplot(aes(fill=Group)) +
  labs(y="Total Fatality Rate", x="BAC", title="BAC vs Fatality")
p2 <- ggplot(sb_law_df,aes(x=factor(SB_Law),y=totrte))+geom_boxplot(aes(fill=Group))+
  labs(y="Total Fatality Rate", x="Seatbelt Laws", title="SB Law vs Fatality")
grid.arrange(p1, p2, nrow = 1)
```

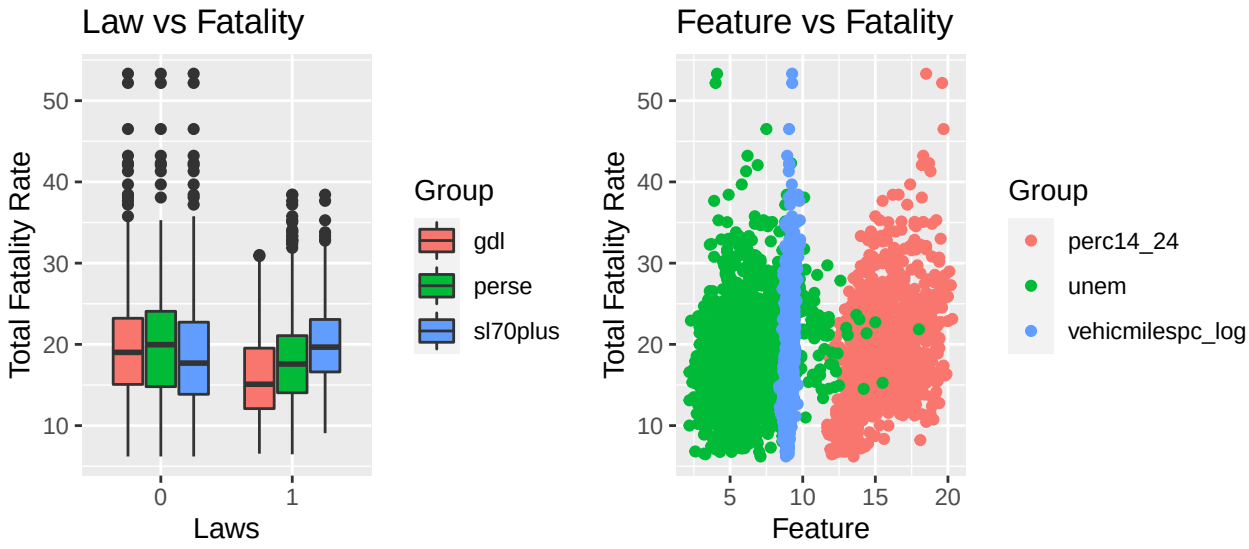


From the box plots, we can see that the average total fatality rate is slightly higher for states without BAC laws compared to states with them. The same interpretation applies to the seatbelt laws. Although the average differences are not very far apart with regards to the total fatality rate, there are more extreme

outlying instances for states without the safety laws.

We can also produce visualizations for another group of laws, which includes `perse` (an administrative license revocation), `gdl` (graduated drivers license law), and `sl70plus` (which is equivalent to if the speed limit == 70 or the speed limit == 75 or the speed limit == NA). Then, we can also examine our non-binary variables together. However, `perc14_24` (the % of the population aged 14 through 24) and `unem` (% of unemployment rate) have a different scale compared to `vehicmilespc` (the vehicle miles traveled in billions) as mentioned earlier. So, we will transform this variable by taking the `log()` of `vehicmilespc` in order to make a proper comparison between these variables.

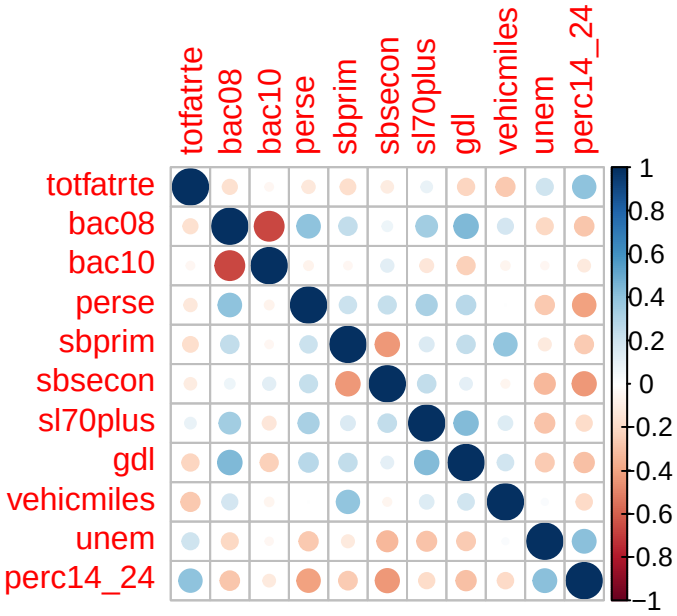
```
all_totfatrte <- c(driving$totfatrte,driving$totfatrte,driving$totfatrte)
all_laws <- c(driving$perse_fac,driving$gdl_fac,driving$sl70plus_fac)
all_group_l <- c(rep("perse",1200),rep("gdl",1200),rep("sl70plus",1200))
law_df <- data.frame(Law=all_laws,totrte=all_totfatrte,Group=all_group_l)
all_fac <- c(driving$perc14_24,driving$unem,log(driving$vehicmilespc))
all_group <- c(rep("perc14_24",1200),rep("unem",1200),rep("vehicmilespc_log",1200))
fac_df <- data.frame(Fac = all_fac, totrte = all_totfatrte, Group = all_group)
p1 <- ggplot(law_df,aes(x=factor(Law),y=totrte)) + geom_boxplot(aes(fill=Group)) +
  labs(y="Total Fatality Rate", x="Laws", title="Law vs Fatality")
p2 <- ggplot(fac_df,aes(x=Fac,y=totrte)) + geom_jitter(aes(col=Group)) +
  labs(y="Total Fatality Rate", x="Feature", title="Feature vs Fatality")
egg::ggarrange(p1, p2, nrow = 1)
```



From the box plot on the left, we can see that the total fatality rate is slightly higher for states without the license laws compared to states with them. Yet, the average fatality rate is actually slightly higher for states with higher/no speed limits even though lower speed limits have a wider fatality distribution. Though, in general, there are more extreme outlying instances for states that do not have safety laws compared to states with them. From our continuous variables, we can see clear groupings from our variables even though they are on the same scale. Yet, no variable has a significantly higher total fatality rate than another. Although there are some outlying values, each variable stays generally below 30% total fatality rate. We can see that there are generally low unemployment rates in the data set. The overall percent of vehicle miles traveled is around 9%. The general percent of the population aged 14 through 24 tends to range from 12% to 20%. This bi-variate analysis provides important insight for the variables used in our model analysis.

We can also examine the correlation between our variables.

```
corrplot::corrplot(cor(driving[ ,c(22,13,12,14,29,30,28,11,25,26,27)]))
```



From the correlation matrix, we can see that some variables are more correlated than others. The two BAC laws are very negatively correlated. The variable that has the strongest positive correlation with `totfatrte` is `perc14-24`. The young population variable is also very positively correlated with the unemployment rate.

After our very thorough EDA, we have a good understanding of the data that we are working with and are able to begin conducting our model analysis.

Question 2

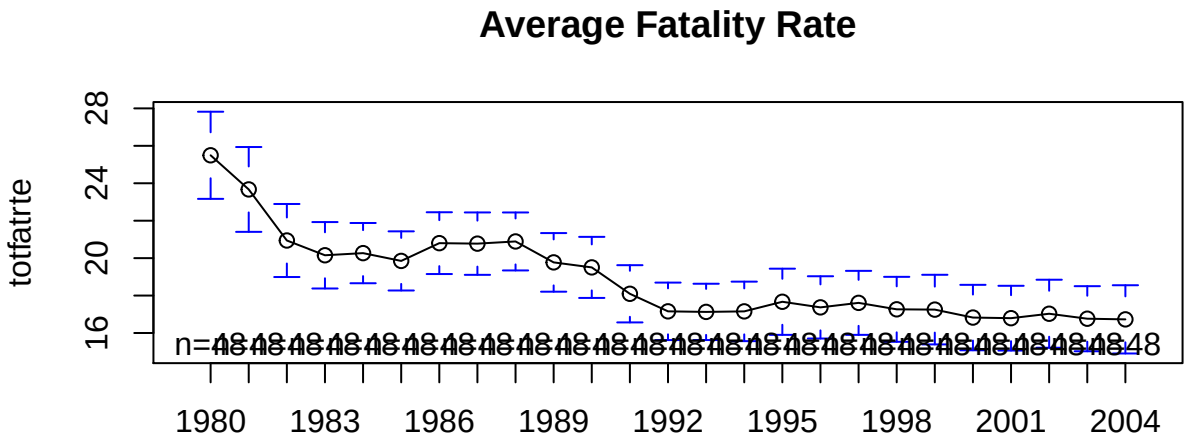
How is the dependent variable of interest `totfatrte` defined? What is the average of this variable in each of the years in the time period covered in this data set? Estimate a linear regression model of `totfatrte` on a set of dummy variables for the years 1981 through 2004. What does this model explain? Describe what you find in this model. Did driving become safer over this period?

The dependent variable `totfatrte` is defined as the total fatalities per 100,000 population. We can find the average of this variable in each of the years in the time period covered in this data set. Then, we can plot these averages in order to visualize the general trend of our dependent variable.

```
avg_per_yr <- aggregate(driving[, "totfatrte"], list(driving$year), mean)
df <- rbind(avg_per_yr$Group.1, avg_per_yr$x)
colnames(df) <- df[1,]
df <- df[-1, ]
df
```

```
##     1980     1981     1982     1983     1984     1985     1986     1987
## 25.49458 23.67021 20.94250 20.15292 20.26750 19.85146 20.80042 20.77479
##     1988     1989     1990     1991     1992     1993     1994     1995
## 20.89167 19.77229 19.50521 18.09479 17.15792 17.12771 17.15521 17.66854
##     1996     1997     1998     1999     2000     2001     2002     2003
## 17.36938 17.61062 17.26542 17.25042 16.82562 16.79271 17.02958 16.76354
##     2004
## 16.72896
```

```
plotmeans(totfatrte ~ year, main="Average Fatality Rate", xlab="", data=driving)
```



In general, there is an overall negative trend as the annual average of the total fatality rate fluctuates, but ultimately declines. The rate starts around 25% then reduces to about 17%.

Then, we can estimate a linear regression model of `totfatrte` on a set of dummy variables for 1981-2004. These variables will include `d81` (which = 1 if year == 1981) to `d04` (which = 1 if year == 2004). First, we need to convert the data into a suitable panel data frame.

```
df.panel <- pdata.frame(driving,index=c("state","year"),drop.index=TRUE,row.names=TRUE)
```

Now, we can build a linear regression model. Through this approach, we will be estimating a pooled-OLS model. This is represented as:

$$y_{i,t} = \beta_0 + \beta_1 * x_{i,t,1} + ... + \beta_k * x_{i,t,k} + \epsilon_{i,t}$$

```
pooled.ols.lm1 <- plm(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, index=c("state", "year"), data=df.panel, model="pooling")
summary(pooled.ols.lm1)
```

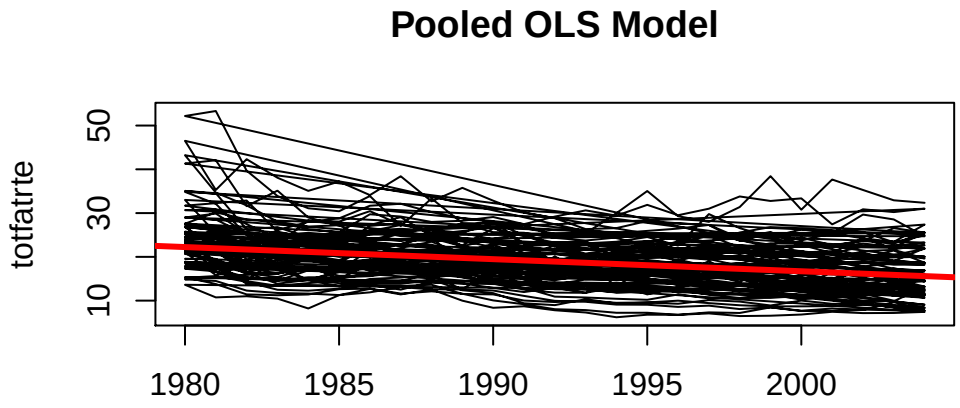
```
## Pooling Model
##
## Call:
## plm(formula = totfatrte ~ d81 + d82 + d83 + d84 + d85 + d86 +
##     d87 + d88 + d89 + d90 + d91 + d92 + d93 + d94 + d95 + d96 +
##     d97 + d98 + d99 + d00 + d01 + d02 + d03 + d04, data = df.panel,
##     model = "pooling", index = c("state", "year"))
##
## Balanced 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
```

Note that the coefficient for each year is relative to 1980, as we could not include a dummy variable for every year, otherwise the dependent variables would be a combination of each other. The negative coefficients we see mean that relative to 1980, fatalities in that year were lower. We see that the coefficients steadily decrease and then level off, signifying that driving deaths decreased and leveled off (as we saw above in our EDA). Almost all of the model coefficients are statistically significant, but the $R^2 = 0.13$ which is very low.

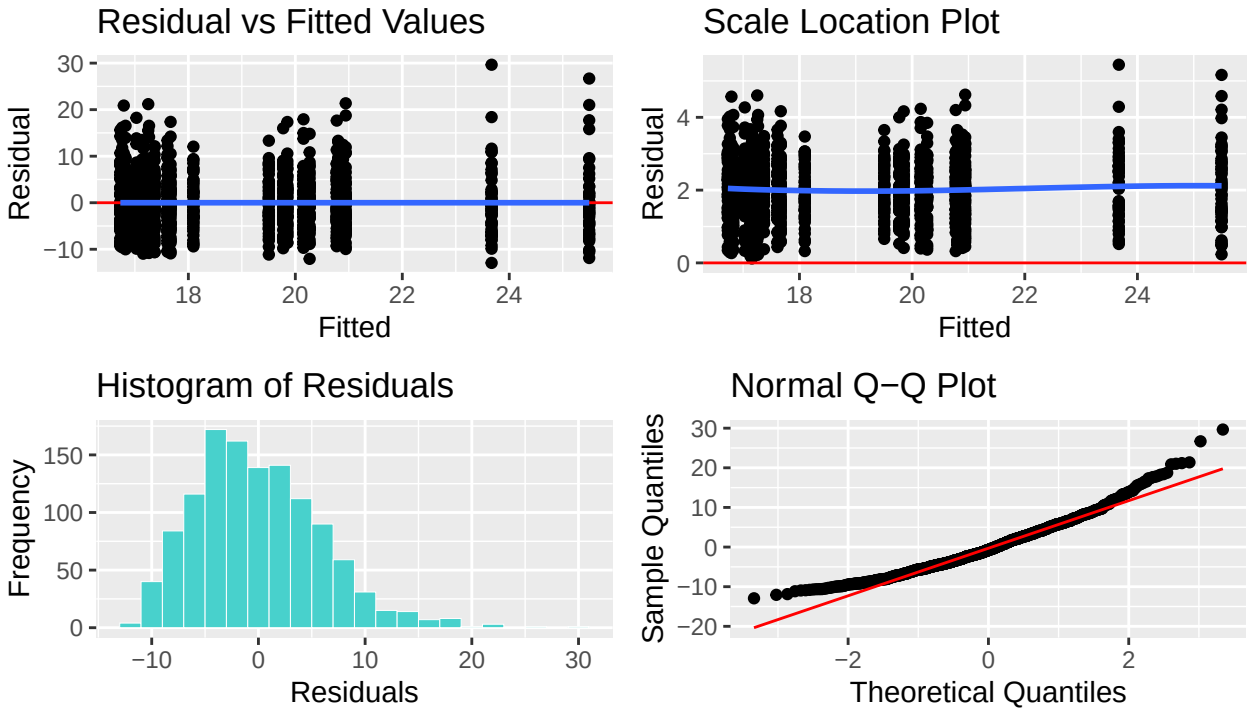
We can also plot our model in order to visualize the regression results.

```
plot(totfatrte ~ year,data=driving,type="l",main="Pooled OLS Model",xlab="")
abline(lm(driving$totfatrte ~ driving$year),lwd=3, col="red")
```



We can see that the model follows the general trend of the data, but does not capture all the variation that occurs across each of the states. Then, we also can conduct some residual diagnostics to asses the fit of the model. We'll write a function to do that here, so that we can apply it again to other models.

```r
residuals_plots <- function(model, driving) {
  dr.res <- resid(model)
  driving$residuals <- dr.res
  driving$index <- seq(1,nrow(driving))
  driving$predicted <- predict(model)
  p1 <- ggplot(driving, aes(x=predicted, y=residuals)) + geom_point() +
    geom_hline(yintercept = 0, color="red") + geom_smooth(method="loess", se=F) +
    labs(y="Residual", x="Fitted", title="Residual vs Fitted Values")
  p2 <- ggplot(driving, aes(x=predicted, y=sqrt(abs(residuals)))) + geom_point() +
    geom_hline(yintercept = 0, color="red") + geom_smooth(method="loess",se=F) +
    labs(y="Residual", x="Fitted", title="Scale Location Plot")
  p3 <- ggplot(driving, aes(residuals)) +
    geom_histogram(binwidth=2, fill="mediumturquoise",col="white",size=0.1) +
    labs(title="Histogram of Residuals",x="Residuals",y="Frequency")
  p4 <- ggplot(driving,aes(sample=residuals))+stat_qq()+stat_qq_line(color="red")+
    labs(title="Normal Q-Q Plot",x="Theoretical Quantiles",y="Sample Quantiles")
  grid.arrange(p1,p2,p3,p4,nrow=2,ncol=2)
}
residuals_plots(pooled.ols.lm1, driving)
```



From the plots we can see that the overall model assumptions are indeed violated due to the panel data format. Although there appears to be zero conditional mean, there is no model linearity and the residuals are not necessarily normally distributed. As a result, this model is not the best fit for our data.

As this model only includes the annual dummy variables, this model explains the effect that each individual year had on the total fatality rate. This annual effect is not restricted to individual states, but reflects the overall impact from that time period. Many things change in a single year, such as a national event or the implementation of a federal law. These types of events are represented by the time variant, but not necessarily the cross-sectional variant. Thus, this model explains the effect of time on the total fatality rate.

So, in this model, we find that each year contributes to the reduction of the total fatality rate. This is because all of the coefficients are negative. Although there is some deviation, the coefficients tend to get larger with

each incremental year as well. This means that the total fatality rate declines more over time. As a result, driving did become safer over this period.

Question 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 carefully explain your rationale, which should be based on your EDA, behind any transformation you made. If no transformation is made, explain why a transformation is not needed. How are the variables bac8 and bac10 defined? Interpret the coefficients on bac8 and bac10. Do per se laws have a negative effect on the fatality rate? What about having a primary seat belt law?

Next, we want to add more explanatory variables to our model. As discovered in our EDA, we want to perform transformations to some of our variables. More specifically, we want to perform a log transformation on vehicmilspc in order to reflect a percent change just like the other non-binary variables. That way, vehicmilspc_log will have the same scale as totfatrte, perc14_24, & unem. The remaining variables are bac08, bac10, perse, sbprim, sbsecon, sl70plus, & gdl. These are all dummy variables, but fractions were recorded if a law was enacted sometime within a year. So, we can round these fractions and transform them to true binary indicator variables for our model.

```
df.panel$vehicmilespc_log <- log(df.panel$vehicmilespc)
```

Now that we have performed our data transformations, we can build our expanded pooled OLS model.

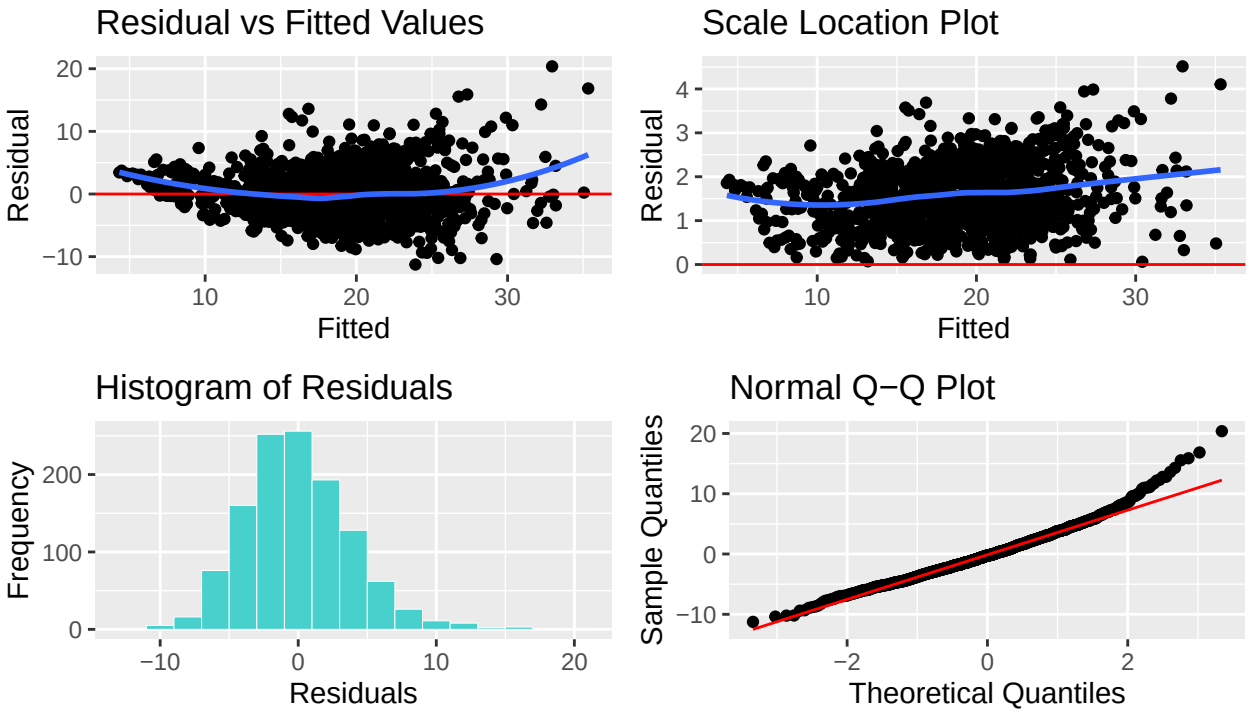
```
pooled.ols.lm2 <- plm(totfatrte ~ d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 + d89 +
  d90 + d91 + d92 + d93 + d94 + d95 + d96 + d97 + d98 + d99 + d00 + d01 + d02 + d03 +
  d04 + bac08_fac + bac10_fac + perse_fac + sbprim_fac + sbsecon_fac + sl70plus_fac +
  gdl_fac + perc14_24 + unem + vehicmilespc_log, data= df.panel,
  model = "pooling", index=c("state", "year"))
summary(pooled.ols.lm2)
```

```
## Pooling Model
##
## Call:
## plm(formula = 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 + bac08_fac +
##     bac10_fac + perse_fac + sbprim_fac + sbsecon_fac + sl70plus_fac +
##     gdl_fac + perc14_24 + unem + vehicmilespc_log, data = df.panel,
##     model = "pooling", index = c("state", "year"))
##
## Balanced Panel: n = 48, T = 25, N = 1200
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -11.25677  -2.62199  -0.27678   2.37313  20.37507
##
## Coefficients:
##                     Estimate  Std. Error  t-value  Pr(>|t|)
## (Intercept)      -233.476461    7.828735 -29.8230 < 2.2e-16 ***
## d81                -2.228750    0.811308  -2.7471  0.006105 **
## d82                -6.943742    0.836999  -8.2960 2.946e-16 ***
## d83                -8.053494    0.849885  -9.4760 < 2.2e-16 ***
## d84                -6.498006    0.855822  -7.5927 6.401e-14 ***
```

```
## d85               -7.249416    0.874672  -8.2882 3.136e-16 ***
## d86               -6.774833    0.912744  -7.4225 2.211e-13 ***
## d87               -7.455702    0.950159  -7.8468 9.609e-15 ***
## d88               -7.781874    0.997444  -7.8018 1.350e-14 ***
## d89               -9.321046    1.036624  -8.9917 < 2.2e-16 ***
## d90              -10.277176    1.061024  -9.6861 < 2.2e-16 ***
## d91              -12.478043    1.084580 -11.5050 < 2.2e-16 ***
## d92              -14.307625    1.106933 -12.9255 < 2.2e-16 ***
## d93              -14.148073    1.119107 -12.6423 < 2.2e-16 ***
## d94              -13.745800    1.138288 -12.0759 < 2.2e-16 ***
## d95              -13.203880    1.165381 -11.3301 < 2.2e-16 ***
## d96              -15.242207    1.205765 -12.6411 < 2.2e-16 ***
## d97              -15.436817    1.222636 -12.6258 < 2.2e-16 ***
## d98              -16.166903    1.241784 -13.0191 < 2.2e-16 ***
## d99              -16.127632    1.254479 -12.8560 < 2.2e-16 ***
## d00              -16.393181    1.273057 -12.8770 < 2.2e-16 ***
## d01              -17.118340    1.291086 -13.2589 < 2.2e-16 ***
## d02              -17.643316    1.304863 -13.5212 < 2.2e-16 ***
## d03              -18.061670    1.310881 -13.7783 < 2.2e-16 ***
## d04              -17.534282    1.342266 -13.0632 < 2.2e-16 ***
## bac08_fac         -2.198018    0.478601  -4.5926 4.854e-06 ***
## bac10_fac         -1.140059    0.354002  -3.2205  0.001315 **
## perse_fac         -0.776130    0.288480  -2.6904  0.007238 **
## sbprim_fac         0.209240    0.481307   0.4347  0.663837
## sbsecon_fac        0.248565    0.420979   0.5904  0.555007
## sl70plus_fac       3.363692    0.424681   7.9205 5.487e-15 ***
## gdl_fac           -0.815273    0.497615  -1.6384  0.101616
## perc14_24          0.151472    0.120121   1.2610  0.207561
## unem               0.821683    0.076527  10.7372 < 2.2e-16 ***
## vehicmilespc_log  28.313373    0.875461  32.3411 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    48612
## Residual Sum of Squares: 18322
## R-Squared:      0.6231
## Adj. R-Squared: 0.6121
## F-statistic: 56.6477 on 34 and 1165 DF, p-value: < 2.22e-16
```

From our results, we can see that our annual variables still steadily decrease and then level off, signifying that driving deaths decreased and leveled off. With regards to our new variables, some are negative, whereas some are positive. This would mean that certain laws (such as high/no speed limit) or features (such as the unemployment rate or the number of miles drive) contribute to higher fatality rates, whereas some safety laws (such as driving sober) help to reduce the fatality rate. Yet once again, many of the coefficients are statistically significant. Now, the $R^2 = 0.62$, which indicates a better model fit than before. Then, we can run residual diagnostics to asses the model fit, using the function above.

```
residuals_plots(pooled.ols.lm2, driving)
```



From the residual plots, we can see that the residuals appear to be normally distributed and the model maintains linearity. However, now, the model does not seem to satisfy the assumption of zero conditional mean. So, although this model might be a better fit than before, it is still not the best for our analysis.

The variable `bac08` is defined as a binary dummy variable that represents whether or not the state blood alcohol limit is 0.08 (where `bac08` = 1 if blood alcohol limit .08). The variable `bac10` is then defined as a binary dummy variable that represents whether or not the state blood alcohol limit is 0.10 (where `bac10` = 1 if blood alcohol limit .10). The coefficient for `bac08_fac` is -2.20 and it is statistically significant. The coefficient for `bac10_fac` is -1.14 and it is also statistically significant. This means that both variables have a statistically significant impact on the reduction of the fatality rate, as they each have a negative slope.

In this model, the "per se laws" are represented by the `perse` variable. It is a binary dummy variable that represents whether or not the state has administrative license revocation (where `perse` = 1 if there are per se laws). The coefficient for `perse_fac` is -0.78 and it is statistically significant. This means that per se laws do have a negative effect on the fatality rate as there is a negative slope. As a result, the per se laws contribute to the reduction of the total fatality rate.

Furthermore, the primary seat belt law is represented by the variable `sbprim` in this model. It is a binary dummy variable that represents whether or not the state has primary seatbelt laws (where `sbprim` = 1 if there are primary seatbelt laws). The coefficient for `sbprim_fac` is 0.21, but it is not statistically significant. Nevertheless, this means that primary seatbelt laws do not have a negative effect on the fatality rate as there is a positive slope. Therefore, primary seatbelt laws do not necessarily contribute to the reduction of the total fatality rate in our model.

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

An alternative way to eliminate the time-invariant unobserved variable is the *fixed effect transformation*. The fixed effect transformation uses the average of an individual over time and the "average equation" from the

original equation. This is represented as:

$$\overline{y_i} = \beta_0 + \beta_1 \overline{x_i} + \overline{a_i} + \overline{\epsilon_i}$$

Each of the cross-sectional subjects' data is demeaned leveraging on time variation in both x and y and, more importantly, the unobserved individual heterogeneity is eliminated within each of the individuals. So, we can re-estimate the model from `Exercise 3` using the fixed effects at the state level and then conduct residual diagnostics to asses the model fit.

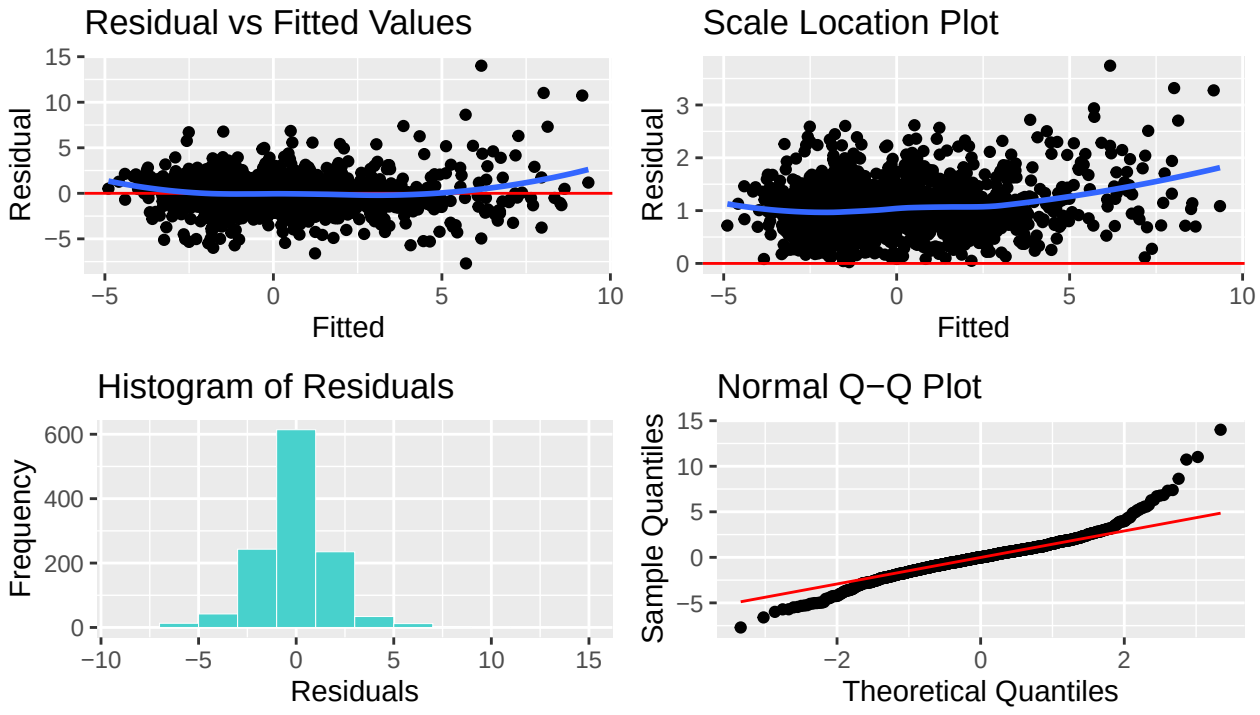
```
plm.fe1 <- plm(totfatrte ~ d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 + d89 + d90 +
  d91 + d92 + d93 + d94 + d95 + d96 + d97 + d98 + d99 + d00 + d01 + d02 + d03 + d04 +
  bac08_fac + bac10_fac + perse_fac + sbprim_fac + sbsecon_fac + sl70plus_fac +
  gdl_fac + perc14_24 + unem + vehicmilespc_log, data = df.panel,
  model = "within", index=c("state", "year"))
summary(plm.fe1)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = 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 + bac08_fac +
##     bac10_fac + perse_fac + sbprim_fac + sbsecon_fac + sl70plus_fac +
##     gdl_fac + perc14_24 + unem + vehicmilespc_log, data = df.panel,
##     model = "within", index = c("state", "year"))
##
## Balanced Panel: n = 48, T = 25, N = 1200
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -7.69945 -0.99544  0.01609  0.96810 14.00898
##
## Coefficients:
##                    Estimate Std. Error  t-value  Pr(>|t|)
## d81               -1.551727   0.407266  -3.8101 0.0001465 ***
## d82               -3.293021   0.437927  -7.5196 1.125e-13 ***
## d83               -4.040870   0.452208  -8.9359 < 2.2e-16 ***
## d84               -4.837155   0.457752 -10.5672 < 2.2e-16 ***
## d85               -5.400893   0.480383 -11.2429 < 2.2e-16 ***
## d86               -4.435107   0.516549  -8.5860 < 2.2e-16 ***
## d87               -5.252093   0.559521  -9.3868 < 2.2e-16 ***
## d88               -5.861995   0.609767  -9.6135 < 2.2e-16 ***
## d89               -7.316287   0.651349 -11.2325 < 2.2e-16 ***
## d90               -7.529423   0.679205 -11.0856 < 2.2e-16 ***
## d91               -8.268807   0.697770 -11.8503 < 2.2e-16 ***
## d92               -9.216108   0.723038 -12.7464 < 2.2e-16 ***
## d93               -9.565049   0.734992 -13.0138 < 2.2e-16 ***
## d94              -10.021712   0.751654 -13.3329 < 2.2e-16 ***
## d95               -9.759143   0.775052 -12.5916 < 2.2e-16 ***
## d96              -10.260365   0.819175 -12.5252 < 2.2e-16 ***
## d97              -10.445305   0.837278 -12.4753 < 2.2e-16 ***
## d98              -11.110527   0.855994 -12.9797 < 2.2e-16 ***
## d99              -11.278462   0.863515 -13.0611 < 2.2e-16 ***
## d00              -11.800592   0.873557 -13.5087 < 2.2e-16 ***
```

```
## d01              -11.559798   0.884453 -13.0700 < 2.2e-16 ***
## d02              -10.914524   0.895641 -12.1863 < 2.2e-16 ***
## d03              -11.019365   0.900744 -12.2336 < 2.2e-16 ***
## d04              -11.375696   0.923360 -12.3199 < 2.2e-16 ***
## bac08_fac         -1.073194   0.324144  -3.3109 0.0009598 ***
## bac10_fac         -0.819497   0.221252  -3.7039 0.0002227 ***
## perse_fac         -1.018550   0.220588  -4.6174 4.335e-06 ***
## sbprim_fac        -1.176187   0.337795  -3.4820 0.0005170 ***
## sbsecon_fac       -0.300895   0.248043  -1.2131 0.2253565
## sl70plus_fac       0.147759   0.255193   0.5790 0.5627007
## gdl_fac           -0.378523   0.275339  -1.3748 0.1694829
## perc14_24          0.194389   0.093362   2.0821 0.0375594 *
## unem              -0.526060   0.060202  -8.7382 < 2.2e-16 ***
## vehicmilespc_log  12.057264   1.156308  10.4274 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    12134
## Residual Sum of Squares: 4404.5
## R-Squared:      0.63702
## Adj. R-Squared: 0.61072
## F-statistic: 57.7071 on 34 and 1118 DF, p-value: < 2.22e-16
```

Once again, many of the coefficients are negative and statistically significant. The $R^2 = 0.64$, which indicates an even better model fit. We can then create the fixed effect model residual plots using the function above.

```
residuals_plots(plm.fe1, driving)
```



From the plots, it appears that the model assumptions are generally satisfied. The residuals seem to be normally distributed, with a slight departure around extreme residual values. There is roughly zero conditional mean and constant variance.

We can also statistically test for serial correlation in our models, using the `pbgtest()` function. In the context of a PBG test, H_0 is that there is no serial correlation of any order up to `p`, with rejection (H_A) indicating that the model does have serial correlation up to order `p`.

```
pbgtest(pooled.ols.lm2, order = 1)
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  totfatrte ~ d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 + d89 +     d90 + d91 + d92 + d93 +
## chisq = 754.4, df = 1, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

```
pbgtest(plm.fe1, order = 1)
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  totfatrte ~ d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 + d89 +     d90 + d91 + d92 + d93 +
## chisq = 257.66, df = 1, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

The resulting p-value for each PBG test is 2.2e-16, which is less than $\alpha = 0.05$. So, we will reject the null hypothesis that each series has no serial correlation of up to order 1 for both of the models. This is an indication that the pooled OLS model and the fixed effects model each have serial correlation in it, which violates both model assumptions.

The coefficients on `bac08, bac10, perse, and sbprim` in the fixed model are all different compared with the pooled OLS estimates. The coefficients for `bac08_fac` and `bac10_fac` increased in the fixed model where the coefficients for `sbprim_fac` and `perse_fac` actually decreased. Also, `sbprim_fac` was positive and not statistically significant for the pooled model, but is negative and statistically significant in the FE model. Overall, in the FE model, each variable contributes a negative effect on the total fatality rate.

From this analysis, we believe that the fixed effect model estimates are still more reliable than the pooled OLS. The pooled OLS standard errors can underestimate true standard errors if they ignore positive serial correlation. Even if ϵ_{it} is uncorrelated, the pooled OLS is likely to be biased and inconsistent when a_i and x_{it} are correlated, which results in heterogeneity bias. As we have seen that each model has serial correlation, we know that this violates model assumptions and the OLS results are unreliable.

The pooled OLS model assumes linearity in parameters. Linearity means that the model relies on variables that have a linear relationship, plus some error. More assumptions are that X has rank $(k+1)$ and that $E(u|X) = 0$, which represents the assumption of zero conditional mean. It also assumes that $Var(u|X) = \sigma^2 I_n$, where I_n is the $n * n$ identity matrix, and $u \sim N(0, \sigma^2 I_n)$, the normality of residuals.

The FE model assumes that the FE estimator is unbiased. So, the error term, ϵ_{it}, is uncorrelated with all the explanatory variables across each period ($E(\epsilon_{it}|x_i, a_i) = 0$ for all i). The model also assumes that the error term is homoscedastic and the error is serially uncorrelated across t.

In this current context, these model assumptions are not reasonable. Although the fixed effects model does not violate the important model assumption of independence like the pooled OLS, both models have serial correlation. This violates the model assumptions and so the results are not necessarily reliable. Therefore, we can explore other types of models as well.

Question 5

Would you prefer to use a random effects model instead of the fixed effects model you built in `Exercise 4`?.

With regards to panel data, a fixed effects model allows for arbitrary correlation between a_i and x_{itj} where the random effects model does not. The FE model is more convincing for estimating ceteris paribus effects. However, if the key explanatory variable is constant over time, a random effects model is more applicable. The RE model assumes that the unobserved effects are uncorrelated with all explanatory variables and requires substantial reasons as to why $Cov(x_{itj}, a_i) = 0$ is reasonable. Yet, the decision between a fixed or random effects model can be determined statistically.

In order to determine whether we should used a random effect model instead of our fixed effect model, we can use the Hausman Test. For this test, the null hypothesis is that the preferred model is random effects vs. the alternative fixed effects model. It tests whether the unique errors u_i are correlated with the explanatory variables; the null hypothesis is they are not. So, we will estimate a random effects model represented as

$$Y_{i,j} = \mu + U_i + W_{i,j}$$

```
plm.re1 <- plm(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 +
  bac08_fac + bac10_fac + perse_fac + sbprim_fac + sbsecon_fac + sl70plus_fac + gdl_fac +
  perc14_24 + unem + vehicmilespc_log, data = df.panel, model = "random",
  index=c("state", "year"))
summary(plm.re1)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = 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 + bac08_fac +
##     bac10_fac + perse_fac + sbprim_fac + sbsecon_fac + sl70plus_fac +
##     gdl_fac + perc14_24 + unem + vehicmilespc_log, data = df.panel,
##     model = "random", index = c("state", "year"))
##
## Balanced Panel: n = 48, T = 25, N = 1200
##
## Effects:
##                  var std.dev share
## idiosyncratic 3.940   1.985 0.314
## individual    8.590   2.931 0.686
## theta: 0.8658
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -7.53769 -1.15534 -0.13187  0.89523 15.30560
##
## Coefficients:
##                      Estimate  Std. Error  z-value  Pr(>|z|)
## (Intercept)      -101.731582   10.062688 -10.1098 < 2.2e-16 ***
## d81                -1.591406    0.419000  -3.7981 0.0001458 ***
## d82                -3.524588    0.449521  -7.8408 4.478e-15 ***
## d83                -4.317536    0.463644  -9.3122 < 2.2e-16 ***
## d84                -5.007727    0.469225 -10.6723 < 2.2e-16 ***
## d85                -5.604597    0.491700 -11.3984 < 2.2e-16 ***
## d86                -4.688896    0.528020  -8.8802 < 2.2e-16 ***
## d87                -5.550690    0.570584  -9.7281 < 2.2e-16 ***
```

```
## d88                -6.190510    0.620645  -9.9743 < 2.2e-16 ***
## d89                -7.680045    0.662126 -11.5991 < 2.2e-16 ***
## d90                -7.960352    0.689597 -11.5435 < 2.2e-16 ***
## d91                -8.781680    0.708238 -12.3993 < 2.2e-16 ***
## d92                -9.809244    0.732950 -13.3832 < 2.2e-16 ***
## d93               -10.140880    0.744939 -13.6130 < 2.2e-16 ***
## d94               -10.565484    0.761777 -13.8695 < 2.2e-16 ***
## d95               -10.308128    0.785213 -13.1278 < 2.2e-16 ***
## d96               -10.889614    0.829325 -13.1307 < 2.2e-16 ***
## d97               -11.093450    0.847170 -13.0947 < 2.2e-16 ***
## d98               -11.786532    0.865578 -13.6170 < 2.2e-16 ***
## d99               -11.962421    0.873010 -13.7025 < 2.2e-16 ***
## d00               -12.480935    0.883278 -14.1302 < 2.2e-16 ***
## d01               -12.319346    0.893688 -13.7848 < 2.2e-16 ***
## d02               -11.761878    0.904268 -13.0071 < 2.2e-16 ***
## d03               -11.890674    0.909257 -13.0774 < 2.2e-16 ***
## d04               -12.216193    0.932340 -13.1027 < 2.2e-16 ***
## bac08_fac          -1.128494    0.330977  -3.4096 0.0006506 ***
## bac10_fac          -0.849730    0.226412  -3.7530 0.0001747 ***
## perse_fac          -0.982930    0.223991  -4.3882 1.143e-05 ***
## sbprim_fac         -1.107767    0.344068  -3.2196 0.0012836 **
## sbsecon_fac        -0.280167    0.254209  -1.1021 0.2704131
## sl70plus_fac        0.263516    0.261338   1.0083 0.3132937
## gdl_fac            -0.376658    0.282682  -1.3324 0.1827147
## perc14_24           0.203128    0.094825   2.1421 0.0321816 *
## unem               -0.446704    0.060988  -7.3245 2.398e-13 ***
## vehicmilespc_log   14.301455    1.121876  12.7478 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    12791
## Residual Sum of Squares: 4859.3
## R-Squared:      0.62011
## Adj. R-Squared: 0.60902
## Chisq: 1901.68 on 34 DF, p-value: < 2.22e-16
```

Now, we can perform the statistical test. If the p-value is significant (e.g. $p \leq 0.05$), then we know that we would prefer to use the fixed effects model. Otherwise, we would prefer to use the random effects model.

```
phtest(plm.fe1, plm.re1)
```

```
##
##  Hausman Test
##
## data:  totfatrte ~ d81 + d82 + d83 + d84 + d85 + d86 + d87 + d88 + d89 +  ...
## chisq = 29.182, df = 34, p-value = 0.7028
## alternative hypothesis: one model is inconsistent
```

The resulting p-value of the Hausman test is 0.70, which is greater than $\alpha = 0.05$. So, we can not reject the null hypothesis that the unique errors u_i are not correlated with the explanatory variables. Therefore, we would prefer to use a random effects model instead of a fixed effects model as the observed explanatory variables and unobserved fixed effects are not correlated.

Question 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`? Please interpret the estimate.

We can use our fixed effect model to determine the estimated effect of changes of different variables on the total fatality rate. This is accomplished by multiplying a change (either positive or negative) on the variable with its coefficient from model.

As we performed a log transformation on the `vehicmilespc` variable in our model (represented as `vehicmilespc_log`), we must also perform a log transformation on the increase of 1,000 miles driven per capita in order to accurately determine the estimated effect.

```
inc_log <- round(log(1000),4)
est_effect <- round(inc_log * plm.fe1$coefficients['vehicmilespc_log'],4)
cat(paste0("The estimated effect on `totfatrte` = ",est_effect," from a ",
             inc_log, "% increase in `vehicmilespc_log`."))
```

```
## The estimated effect on `totfatrte` = 83.2892 from a 6.9078% increase in `vehicmilespc_log`.
```

As a result, a 6.9% increase of miles driven per capita corresponds to a 83.23 point estimate increase on the total fatality rate. We can also compute the confidence interval for this estimated effect.

```
se <- sqrt(diag(vcov(plm.fe1)))[34]
z <- qnorm((1+0.95)/2)
ctab <- cbind(est_effect-z*se,est_effect+z*se)
colnames(ctab) <- stats:::format.perc(c((1-0.95)/2,(1+0.95)/2),digits=3)
ctab
```

```
##                      2.5 %   97.5 %
## vehicmilespc_log 81.02288 85.55552
```

Therefore, the estimated effect on the total fatality rate is 83.23, but it can range from 81.64 to 86.17 given a 6.9% increase in the percent of miles driven.

Question 7

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

Panel data (or longitudinal data) have both cross-sections and time series dimensions as it samples the same group over time. It records characteristics and responses over multiple time points. This helps to understand dynamic behavior and how things are related to other variables. However, there is serial correlation in this data as responses on one occasion provides information about the likely value of the response on a future occasion. This is known as heterogeneous variability, which is the variance of the response as it changes over the duration of the study. Therefore, we can employ panel-specific models too account for this effect.

Yet, it is still possible for a panel-model to have serial correlation or heteroskedasticity in the idiosyncratic errors of the model. If this is the case, this quality violates the model assumptions of independence and homogeneity of variance. In other words, the independent and identically distributed (iid) data assumption is not satisfied due to the repeated observations, making the standard errors and test statistics incorrect and invalid. As a consequence, these violations lead to unreliable estimates with incorrect statistical inferences.