Drugs and Jobs: The Effect of Unemployment on Drug Overdose Deaths in America

Evan Arnold and Caleb Ren

12/06/2019

Introduction and Motivation

Overdose deaths in the US have increased dramatically since 1999. Opioid use deaths in particular have reached epidemic levels, with a 200% increase in overdose deaths between 2000 and 2014 (Ghertner and Groves 2018). The Centers for Disease Control and Prevention estimates that over 60,000 drug overdose deaths occurred in 2016, three times the rate of drug deaths as 1999 (Hedegaard and Warner 2018)

Our analysis has two goals. First, we will determine the significane of unemployment when predicting overdose death rates, and validate this significance in the presence of other predictors. Second, we will determine which model from a set of possible, well-tuned models has the most predictive ability.

Exploratory Data Anslysis Methods

Data Summary

As a next step in our project, we collected data from the CDC in the form of the Vital Statistics Rapid Release dataset (VSRR). The VSRR data contains provisional counts of drug overdose deaths in the US as reported by agencies from all 50 states and the District of Columbia. The data is collected on a monthly basis.

The data of import to this project is the number of deaths in each state as a result of drug overdose. Drug overdoses are counted by state agencies in acordance to World Health Organization standards, which lay out the basic guides for reporting agencies to code and classify causes of death. Drug categories that are represented in this dataset include the major drivers of the opioid epidemic like heroin (coded by T40.1), natural opioid analgesics (morphine and codeine), synthetic opioids (oxycodone, hydrocodone, oxymorphone; T40.2), methadone (T40.3), other synthetics (fentanyl, tramadol; T40.4) and other drugs like cocaine, methamphetamine, etc.

There were over 26052 data points from the VSRR dataset. Of those data points, many are individual observation of different coded deaths from different drugs; after reshaping and data cleaning, there are now 2652 individual observations. The data ranges from Januart 1, 2015 to April 1, 2019, with each state reporting 52 observations (once per month). Overdose deaths range from 55 deaths in the month of May 2015 in South Dakota to a high of 5697 in Pennsylvania in September of 2017.

```
# overdose dataset
overdose <- read.csv("data/overdose.csv")

# remove columns
bad.cols <- c("Period", "Percent.Complete", "Percent.Pending.Investigation",
              "State.Name", "Footnote", "Footnote.Symbol", "Predicted.Value")
overdose <- overdose[,!(colnames(overdose) %in% bad.cols)]

# reshape dataframe to wide format
overdose <- reshape(overdose, idvar = c("State", "Year", "Month"),
```

```
                    timevar = "Indicator", direction = "wide")

# proper column naames
names <- c("state", "year", "month", "overdoseDeaths",
           "natural.semiSynthetic.synthetic.methadone",
           "opioids", "cocaine", "stimulants", "deaths",
           "synthetic.noMethadone", "heroin",
           "natural.semiSynthetic.methadone",
           "natural.semiSynthetic", "percentSpecified",
           "methadone")
colnames(overdose) <- names

# remove aggregate statistics
overdose <- overdose[!(overdose$state %in% c("US", "YC")),]
overdose$state <- droplevels(overdose$state)

# reformat month to ordered factor
months.levels <- c("January", "February", "March", "April", "May", "June",
                   "July", "August", "September", "October", "November", "December")
months.labels <- substr(tolower(months.levels), 1, 3)
overdose$month <- ordered(overdose$month, levels = months.levels, labels = months.labels)

# relevant columns
overdose <- overdose[,1:4]

# convert year to factor
overdose$year <- factor(overdose$year)

# adding dates
overdose$dates <- as.Date(paste("01", overdose$month, overdose$year, sep =""),
                          format = "%d%b%Y")
```

Unemployment data was sourced from the Bureau of Labor Statistics. Unemployment data is published in monthly increments from the Bureau of Labor Statistics by state. Data is published beginning in 1976 and is published on the first of each month describing the previous month's unemployment rate.

There is a very specific definition of who in the labor force is considered *unemployed*. According to the BLS, those who are currently unemployed are those who are "jobless, looking for a job, and avaiable for work." People who are incarcerated, in a nursing home, or in a mental health care facility are not considered unemployed as they are not fit for work.

Using this definition, data was scraped from the BLS website and aggregated by each state and the District of Columbia. The unemployment rate in percent is given by the `unemployment` column. The lowest unemployment rate in a given state and month is Vermont in 2019 with a 2.1% unemployment rate. The highest rate is DC in 2015 with a 7.4% unemployment rate. The data itself is roughly Normally distributed with a mean of 4.2% and a median of 4.31%.

We hypothesize that unemployment will have a significant positive association with overdose death rate. We believe unemployment worses community conditions and potentially leads to increase abuse of opioids.

```
# iterate through state data files
unemployment <- data.frame()
for (file in list.files("data/state", full.names = T)) {

  # state name and data
  state <- substr(basename(file), 1, 2)
```

```
  data <- read.csv(file)
  data$state <- rep(state, nrow(data))

  # year and month
  data$year <- sapply(data$DATE, function(x) as.numeric(substr(x, 1, 4)))
  data <- data[data$year >= 2015,]
  month <- sapply(data$DATE, function(x) as.numeric(substr(x, 6, 7)))
  month <- ordered(month, labels = months.labels)
  data$month <- month
  colnames(data)[2] <- "unemployment"

  # record state data
  unemployment <- rbind(unemployment, data)
}
colnames(unemployment)[2] <- "unemployment"
unemployment <- unemployment[,-1] # drop default date column

# merge datasets
overdose <- merge(overdose, unemployment, by = c("state", "year", "month"))

# order
overdose <- overdose[order(overdose$year, overdose$state, overdose$month),]
```

Here we consider a series of variables from the Federal Reserve Bank of St. Louis (Federal Reserve Economic Data or FRED for short). All such datasets are originally from Bureau of Labor Statistics or the U.S. Census Bureau, however, FRED has their data in a more convenient format.

First, we consider the number of new private housing units authorized by building permit. This timeseries is reported monthly. In our analysis, this variable serves as a proxy to housing development as well as the health of the housing market; more permits implies a healthy housing market and ample housing options. We thus expect a negative association between overdose deaths and new housing permits.

Next, we consider the money spent on imports of manufactored and non-manufactored commodities in millions of dollars. This timeseries is reported monthly. This feature will offer our analysis an insight into the health of the state-local manufactoring sector. In the past few years, the states with some of the worst opioid overdose numbers are those in in the Rust Belt (https://www.drugabuse.gov/drugs-abuse/opioids/opioid-summaries-by-state). One potential narrative is that the increased use of overseas labor stips entire communities of their jobs, and many of those effected turn to opioids. We can reasonably quantify this effect by state with the number of dollars spent on imports. We expect an increase in spending on imports to imply a decrease in availability of manufactoring jobs and the health of the local manufactoring sector, and thus we expect a positive association between import spending and overdose deaths.

Now, we consider the value of manufactored and non-manufactored exports in millions of dollars. This timeseries is reported monthly. We hope to use this variable as another perspective on the health of manufactoring and the availability of jobs in the manufactoring sector. We expect the value of exports to have a negative association with overdose deaths.

Finally, we consider per capita personal income in dollars. This timeseries is reported annually. It is reasonable to assume that the average personal income does not change dramatically intrayear. Per capita income affords us a glimpse into the personal financial freedom of state residents. We chose to include per capita income instead of median household income in order to capture the influence of the wealthly. We expect a negative association between income and overdose deaths.

```
# convert Federal Reserve Bank of St. Louis data to long-form
stlouis <- function(wide, var.name, state_start, state_end) {
  # states to include
```

```
  states.include <- levels(overdose$state)

  # state names
  cols.states <- substr(colnames(wide)[-1], state_start, state_end)
  colnames(wide) <- c("date", cols.states)

  # filter for states
  wide <- wide[,c("date", states.include)]

  # expand date column
  wide$month <- ordered(months(as.POSIXct(wide$date)),
                        levels = months.levels,
                        labels = months.labels)
  wide$year <- as.numeric(substr(wide$date, 1, 4))
  wide <- wide[,-1] # remove original date variable
  wide <- wide[,c("year", "month", states.include)]

  # filter years
  wide <- wide[wide$year >= 2015 & wide$year <= 2019,]

  # filter months in 2019 (april is the last month in the overdose dataset in 2019)
  wide <- wide[!(wide$month > "apr" & wide$year == 2019),]

  # convert to longform data
  long <- reshape(wide, direction = "long",
                  varying = states.include,
                  v.names = var.name,
                  idvar = c("year", "month"),
                  times = states.include)

  # column name and order
  colnames(long)[3] <- "state"
  long <- long[,c("state", "year", "month", var.name)]

  return(long)
}

# read in data
imports <- read.table("data/imports/Imports.txt", header = T)
permits <- read.table("data/permits/Permits.txt", header = T)
income <- read.table("data/income/income.txt", header = T)

# handle export data (multi-file dataset)
multmerge <- function(mypath) {
  filesnames <- list.files(path = "data/exports", full.names = T)
  datalist <- lapply(filesnames, function(x) {read.csv(file = x, header = T) })
  Reduce(function(d1, d2) merge(d1, d2, by = "DATE"), datalist)
}
exports <- multmerge("data/exports")
names(exports) <- c("dates", substr(names(exports)[-1], 7, 9))

# convert to long-form
permits <- stlouis(permits, "permits", 1, 2)
```

```r
imports <- stlouis(imports, "imports", 7, 8)
income <- stlouis(income, "income", 1, 2)[-3]
exports <- reshape(exports,
                   direction = "long", varying = names(exports)[-1],
                   v.names = "exports", timevar = "state",
                   times = names(exports)[-1])
exports$dates <- as.Date(exports$date)
exports$id <- NULL

# combine with overdose and unemployment data
overdose <- merge(overdose, permits, by = c("state", "year", "month"))
overdose <- merge(overdose, imports, by = c("state", "year", "month"))
overdose <- merge(overdose, exports, by = c("state", "dates"))
overdose <- merge(overdose, income, by = c("state", "year"))
```

Below, we include annual population estimates from the Census Bureau. We use annual data for two reasons: the availability of state population estimates is limited and we can reasonably assume that state population does not change dramatically intrayear. Furthermore, we use the population estimate from the previous year as a given year's population variable. We do so becuase population estimates are generated late into the year, and thus for any given year, the previous year's Census Bureau estimate is likely more accurate. We will use population to normalize the other predictors.

```r
# data
population <- read.csv("data/population.csv")

# filter for 50 states and DC
population <- population[6:(nrow(population) - 1),]

# state names
abbrev <- function(state) {
  if (state == "District of Columbia") {
    return("DC")
  }
  return(state.abb[which(state.name == state)])
}
population$state <- unlist(sapply(population$NAME, abbrev))

# relevant variables
population <- population[,c("state", paste(rep("POPESTIMATE", 5),
                                           2014:2018, sep = ""))]

# columns names
pop.cols <- as.character(2015:2019)
colnames(population) <- c("state", pop.cols)

# convert to long format
population <- reshape(population, direction = "long",
                      varying = pop.cols,
                      v.names = "population",
                      times = pop.cols)
population <- population[,-4]
colnames(population)[2] <- "year"

# merge with overdose dataset
```

```r
overdose <- merge(overdose, population, by = c("state", "year"))
```

R inherently has several statistics about states. Most of these are static estimates from the early 70's (population estimates, income per capita, illiteracy,...). We can, however, use the datasets which give information about the region a state is in. This is a categorical variable with four categories: northeast, northcentral, south, and west. Though the nomenclature is dated, these categories reasonably divide states by potentially important factors. For example, we expect the Rust Belt (which R includes in the region northcentral) to have a positive association with overdose deaths (as noted above). Here we note that we have to encode Washington DC ourselves as the data in R does not contain the District of Columbia (in accordance with R's encoding, Washington DC is in the south).

R also provides us with the area of each state in square miles. Again, we must encode Washington DC manually (source: https://www.britannica.com/place/Washington-DC).

```r
# region and area
overdose$region <- rep(NA, nrow(overdose))
overdose$area <- rep(NA, nrow(overdose))
for (i in 1:length(state.region)) {
  state.row <- overdose$state == state.abb[i]
  overdose$region[state.row] <- as.character(state.region[i])
  overdose$area[state.row] <- state.area[i]
}
overdose$region[overdose$state == "DC"] <- "South"
overdose$area[overdose$state == "DC"] <- 68
overdose$region <- factor(overdose$region)
```

Before we begin any modeling or analyis, we must normalize overdose deaths, permits, imports, and exports. By doing so, we convert each to a rate which can be directly compared across states. In order to avoid working with very small numbers, we convert each rate to: overdose deaths per 100000 people, permits per 100000 people, spending on imports in millions of dollars per 100000 people, and value of exports in millions of dollars per 100000 people.

```r
# normalize raw predictors
overdose$overdoseDeaths <- (overdose$overdoseDeaths / overdose$population) * 100000
overdose$permits <- (overdose$permits / overdose$population) * 100000
overdose$imports <- (overdose$imports / overdose$population) * 100000
overdose$exports <- (overdose$exports / overdose$population) * 100000
```

EDA

Below we check to see if any cells of the dataframe are NA, NULL, NaN, or infinite. No such values exist. We have a relatively clean dataset.

```r
# number of empty cells
finite <- apply(overdose, 2, function(x) any(is.infinite(x)))
nans <- apply(overdose, 2, function(x) any(is.nan(x)))
nas <- is.na(overdose)
nulls <- is.null(overdose)
any(finite | nans | nas | nulls)
```

```
## [1] FALSE
```

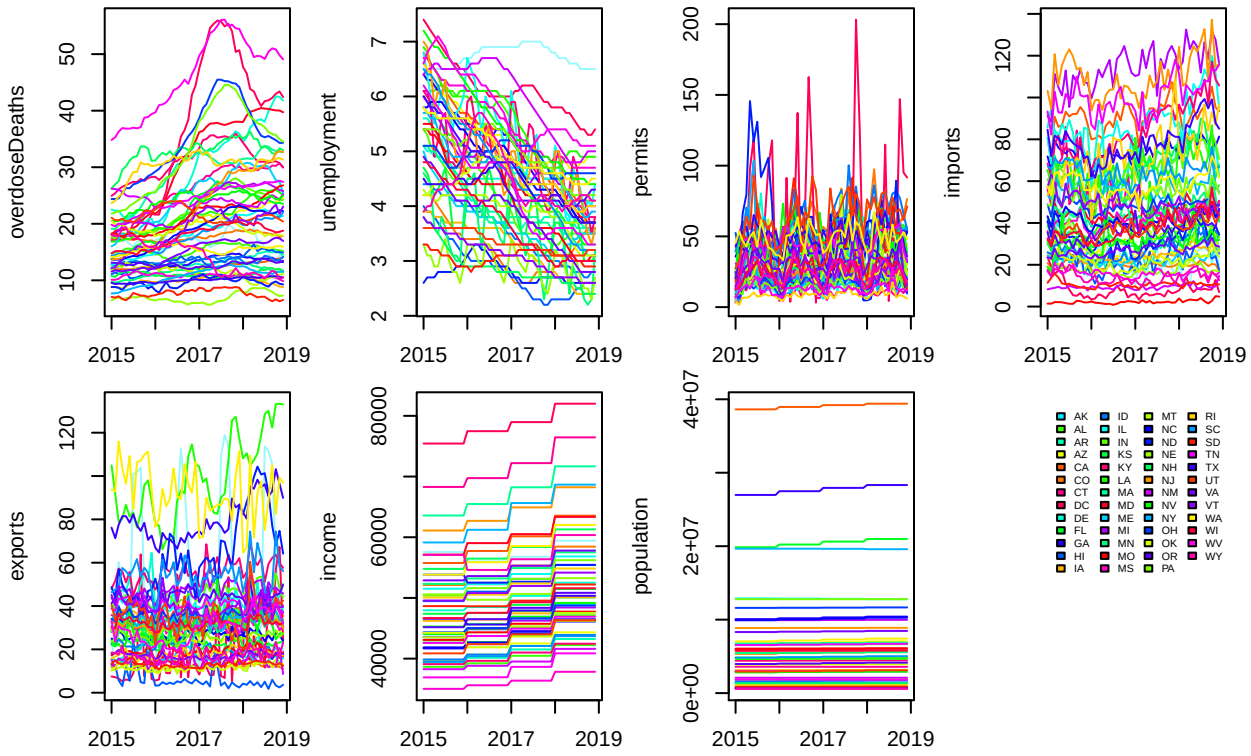
We now consider the possibility of multicollinearity. As shown below, some of the quantitative features of our dataset are highly correlated (population and imports, imports and exports,...). In order to handle this multicolinearity, we will consider ensemble models such as random forests.

```r
round(cor(overdose[,c(6:11, 13)]), 3)
```

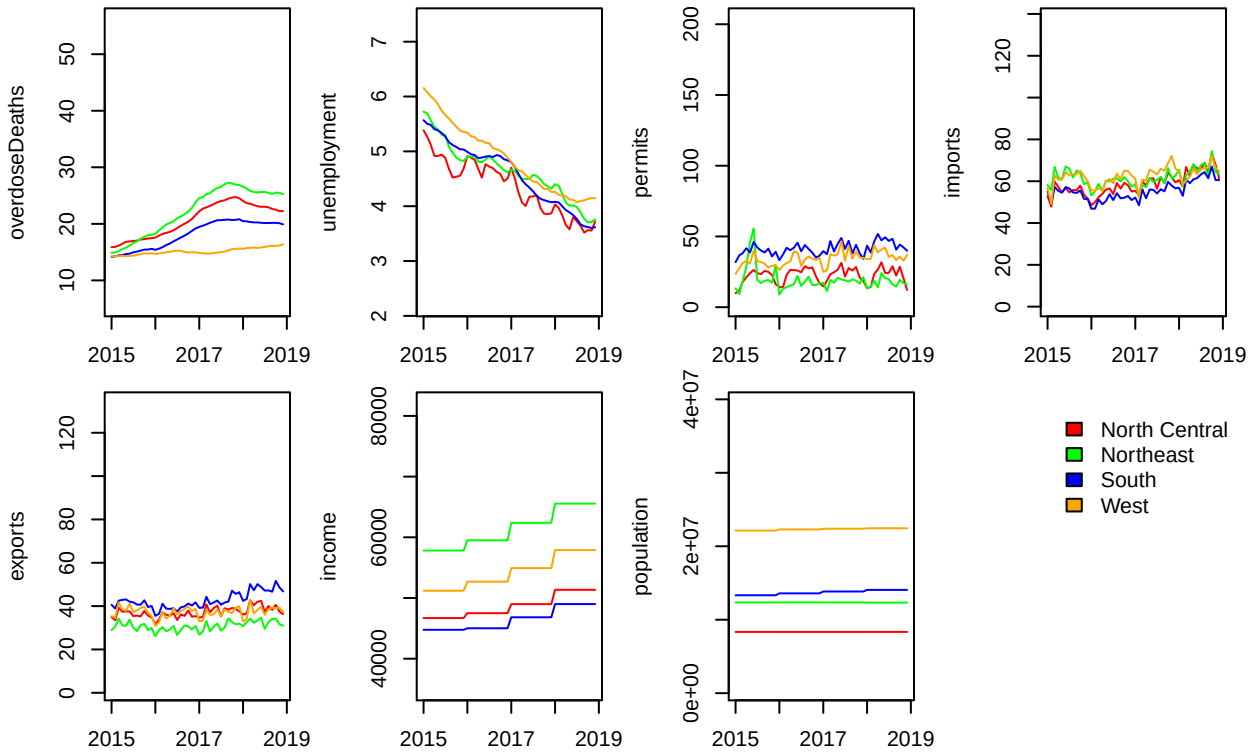
```
##              unemployment permits imports exports income population   area
## unemployment        1.000  -0.197   0.092   0.142 -0.087      0.151  0.254
## permits            -0.197   1.000  -0.088   0.104 -0.054     -0.025  0.030
## imports             0.092  -0.088   1.000   0.447  0.072      0.433 -0.133
## exports             0.142   0.104   0.447   1.000 -0.022      0.192  0.191
## income             -0.087  -0.054   0.072  -0.022  1.000      0.121 -0.041
## population          0.151  -0.025   0.433   0.192  0.121      1.000  0.150
## area                0.254   0.030  -0.133   0.191 -0.041      0.150  1.000
```

```r
states <- levels(overdose$state)
regions <- levels(overdose$region)
col_state <- sample(rainbow(51))
col_reg <- c("red", "green", "blue", "orange")
plot_state <- function(v) {
  mydf <- subset(overdose, state == "AK")
  plot(mydf[,v][order(mydf$dates)] ~ mydf$dates[order(mydf$dates)],
       type = "l",
       ylim = range(overdose[,v]),
       col = alpha(col_state[1], 0.4), xlab = "", ylab = v)
  for (i in 2:(length(states))) {
    mydf <- subset(overdose, state == states[i])
    lines(mydf[,v][order(mydf$dates)] ~ mydf$dates[order(mydf$dates)],
          col = col_state[i])
  }
}


plot_region <- function(v) {
  mydf <- subset(overdose, region == regions[1])
  temp <- unlist(by(mydf, mydf$dates,
              FUN = function(x) {weighted.mean(x[,v], x[,"population"])},
              simplify =F))
  plot(temp ~ as.Date(names(temp)),
       type = "l",
       ylim = range(overdose[,v]),
       col = col_reg[1], xlab = "", ylab = v)
  for (i in 2:(length(regions))) {
    mydf <- subset(overdose, region == regions[i])
    temp <- unlist(by(mydf, mydf$dates,
              FUN = function(x) {weighted.mean(x[,v], x[,"population"])},
              simplify =F))
    lines(temp ~ as.Date(names(temp)),
          col = col_reg[i])
  }
}

par(mfrow = c(2, 4), mar = c(2, 4, 1, 1))
for (v in c("overdoseDeaths", "unemployment", "permits", "imports",
            "exports", "income", "population")) {
  plot_state(v)
}
plot(NULL ,xaxt='n',yaxt='n',bty='n',ylab='',xlab='', xlim=0:1, ylim=0:1)
```

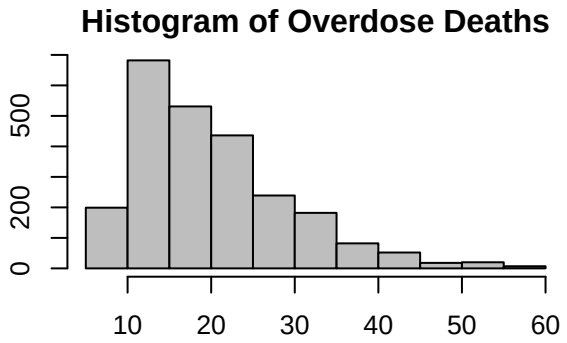
```
legend(0, 1, legend = states, cex=0.5, bty='n',
    fill = col_state, ncol = 4)
```

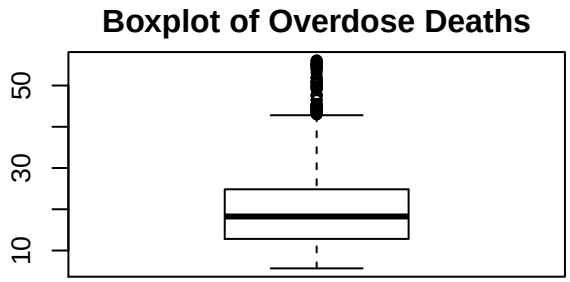


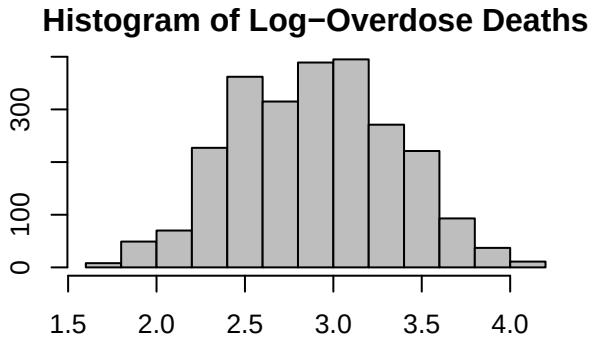

```
par(mfrow = c(2, 4), mar = c(2, 4, 1, 1))
for (v in c("overdoseDeaths", "unemployment", "permits", "imports",
            "exports", "income", "population")) {
  plot_region(v)
}
plot(NULL ,xaxt='n',yaxt='n',bty='n',ylab='',xlab='', xlim=0:1, ylim=0:1)
legend(0, 1, legend =regions, cex=1, bty='n',
    fill = col_reg)
```

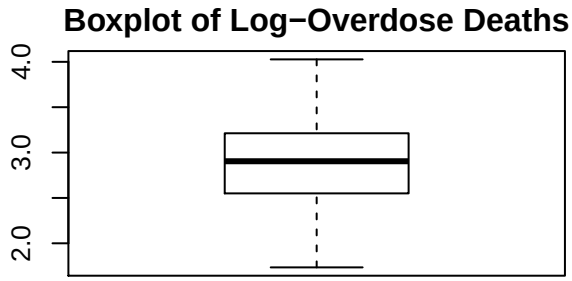



```
# histogram and boxplot of response
par(mfrow = c(2, 2), mar = c(3, 2, 2, 2))
hist(overdose$overdoseDeaths, main = "Histogram of Overdose Deaths",
     xlab = "Overdose Deaths", col = "grey")
boxplot(overdose$overdoseDeaths, main = "Boxplot of Overdose Deaths",
        xlab = "Overdose Deaths")

# histogram and boxplot of log-response
hist(log(overdose$overdoseDeaths), main = "Histogram of Log-Overdose Deaths",
     xlab = "Log-Overdose Deaths", col = "grey")
boxplot(log(overdose$overdoseDeaths), main = "Boxplot of Log-Overdose Deaths",
        xlab = "Log-Overdose Deaths")
```



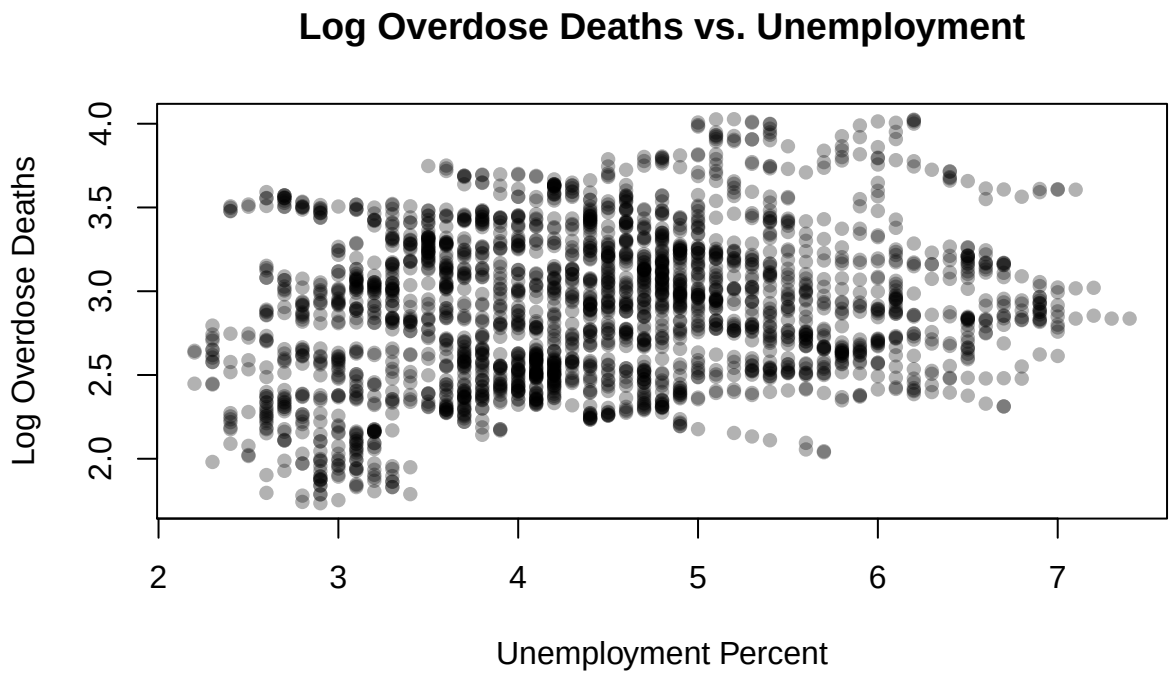






We see that the data is much closer to a Normal distribution if we apply a log transformation. We shall predict the log-rate of overdose deaths in our models.

```
# response vs. unemployment
par(mfrow = c(1, 1))
plot(log(overdoseDeaths) ~ unemployment, data = overdose,
     main = "Log Overdose Deaths vs. Unemployment",
     xlab = "Unemployment Percent", ylab = "Log Overdose Deaths",
     pch = 16, col = rgb(0, 0, 0, 0.3))
```



Baseline Model

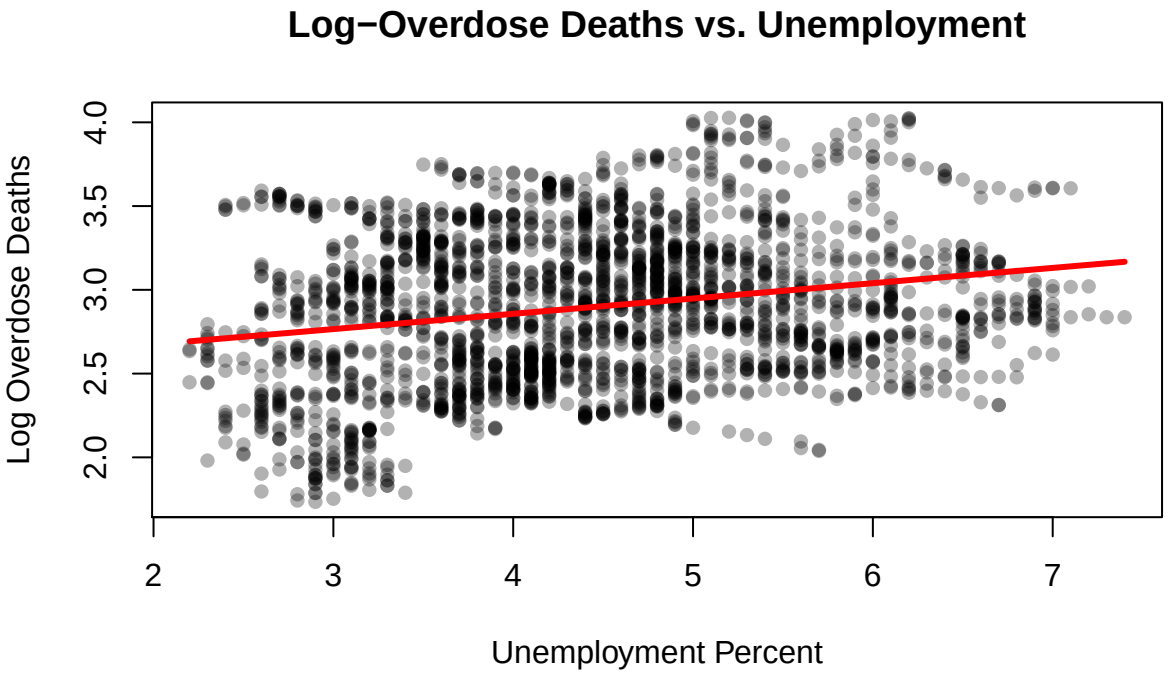
```
# train/test data for cross validation
samples <- c()
for (i in 1:length(states)) {
  lower <- (1 + (i - 1) * 48); upper <- (48 * i)
  samples <- c(samples, sample(lower:upper, size = 38))
}
train <- overdose[samples,]
test <- overdose[-samples,]

# rmse function
rmse <- function(m, o){
  return(sqrt(mean((m - o)^2)))
}

# simple linear model
summary(lm1 <- lm(log(overdoseDeaths) ~ unemployment, data = train))$coefficients[2,]
```

```
##     Estimate   Std. Error      t value     Pr(>|t|)
## 9.132573e-02 9.067219e-03 1.007208e+01 2.719915e-23
```

```
lm1.train <- rmse(predict(lm1, train), log(train$overdoseDeaths))
lm1.test <- rmse(predict(lm1, test), log(test$overdoseDeaths))
```

The simple regression model has a positive coefficient for unemployment (0.09179). With a t-statistic of 10.11 (p-value ≈ 0), this coefficient is very significant. The model has a positive association between unemployment and overdose deaths.

```
# observations with simple regression line
x <- seq(min(overdose$unemployment), max(overdose$unemployment), 0.01)
y <- predict(lm1, newdata = data.frame(unemployment = x))
plot(log(overdoseDeaths) ~ unemployment, data = overdose,
     main = "Log-Overdose Deaths vs. Unemployment",
     xlab = "Unemployment Percent", ylab = "Log Overdose Deaths",
     col = rgb(0, 0, 0, 0.3), pch = 16)
lines(y ~ x, col = "red", lwd = 3)
```



In order to better evaluate the importance of unemployment, we apply an ESS F-test. First we fit two linear models. The first model include all main effects except unemployment, while the second includes all main effects. Then, we fit two quadratic models. The first model contains all main effects and their respective quadratic variants except unemployment. The second model contains the same predictors, except that it includes unemployment and its quadratic effect. We again verify the assumptions of a linear model via a plot of residuals vs. fitten values.

```
# linear model with all main effects excluding unemployment
lm2 <- lm(log(overdoseDeaths) ~ state + month + year +
            permits + imports + exports + income +
            population + region + area,
          data = train)

# linear model with all main effects including unemployment
lm3 <- lm(log(overdoseDeaths) ~ state + month  + year +
            permits + imports + exports + income +
            population + region + area + unemployment,
          data = train)

# ESS F-test
anova(lm2, lm3)
```

```
## Analysis of Variance Table
##
## Model 1: log(overdoseDeaths) ~ state + month + year + permits + imports +
##     exports + income + population + region + area
## Model 2: log(overdoseDeaths) ~ state + month + year + permits + imports +
##     exports + income + population + region + area + unemployment
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1868 22.807
## 2   1867 22.634  1   0.17353 14.314 0.0001596 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# relevant coefficients
coef(lm3)[c("unemployment", "income", "imports", "exports", "permits")]
```

```
##  unemployment        income       imports       exports       permits
## -2.949942e-02  3.903526e-05 -8.922224e-04  7.284677e-04 -1.773442e-04
```

```
# record predictive results
lm2.train <- rmse(predict(lm2, train), log(train$overdoseDeaths))
lm2.test <- rmse(predict(lm2, test), log(test$overdoseDeaths))
lm3.train <- rmse(predict(lm3, train), log(train$overdoseDeaths))
lm3.test <- rmse(predict(lm3, test), log(test$overdoseDeaths))
```

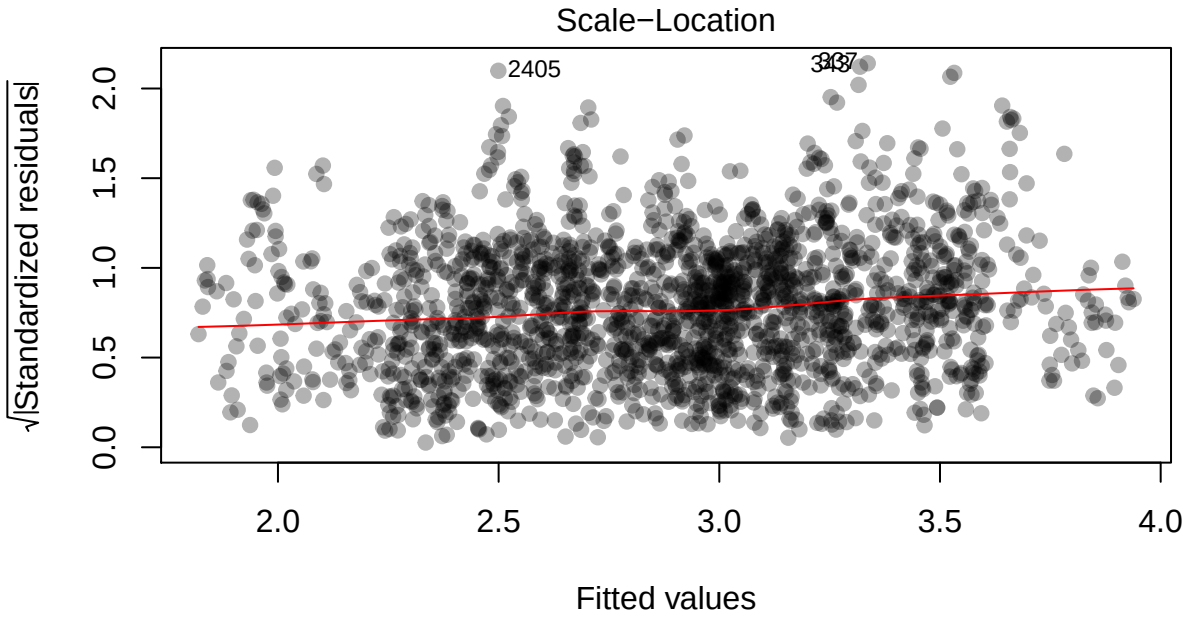
With an F-statistic of 13.387 (with 1887, 1 degrees of freedom) and a corresponding p-value < 0.001, unemployment provides significant predictive ability. We now consider the same ESS F-test with quadratic models.

Unemployment now has a negative coefficient. This is likely due to multicollinearity following from the results of our simple linear model. Regardless, this result contradicts our hypothesis concerning unemployment. We do, however, see positive coefficients for income and exports, and a negative coefficient for imports (as we hypothesized in data collection and EDA). The coefficient for permits is negative, which contradicts our hypothesis.

```
# quadratic model with all main effects excluding unemployment
polynomial1 <- lm(log(overdoseDeaths) ~ state + month  + year +
                  poly(permits, 2, raw = T) + poly(imports, 2, raw = T)
                  + poly(exports, 2, raw = T) + poly(income, 2, raw = T)
                  + poly(population, 2, raw = T) + region + area,
                  data = train)

# quadratic model with all main effects including unemployment
polynomial2 <- lm(log(overdoseDeaths) ~ state + month  + year +
                  poly(permits, 2, raw = T) + poly(imports, 2, raw = T)
                  + poly(exports, 2, raw = T) + poly(income, 2, raw = T)
                  + poly(population, 2, raw = T) + region + area
                  + poly(unemployment, 2, raw = T),
                  data = train)

# ESS F-test
anova(polynomial1, polynomial2)
```

```
## Analysis of Variance Table
##
## Model 1: log(overdoseDeaths) ~ state + month + year + poly(permits, 2,
##     raw = T) + poly(imports, 2, raw = T) + poly(exports, 2, raw = T) +
##     poly(income, 2, raw = T) + poly(population, 2, raw = T) +
##     region + area
## Model 2: log(overdoseDeaths) ~ state + month + year + poly(permits, 2,
##     raw = T) + poly(imports, 2, raw = T) + poly(exports, 2, raw = T) +
##     poly(income, 2, raw = T) + poly(population, 2, raw = T) +
##     region + area + poly(unemployment, 2, raw = T)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1863 22.145
## 2   1861 21.816  2   0.32939 14.049 8.792e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

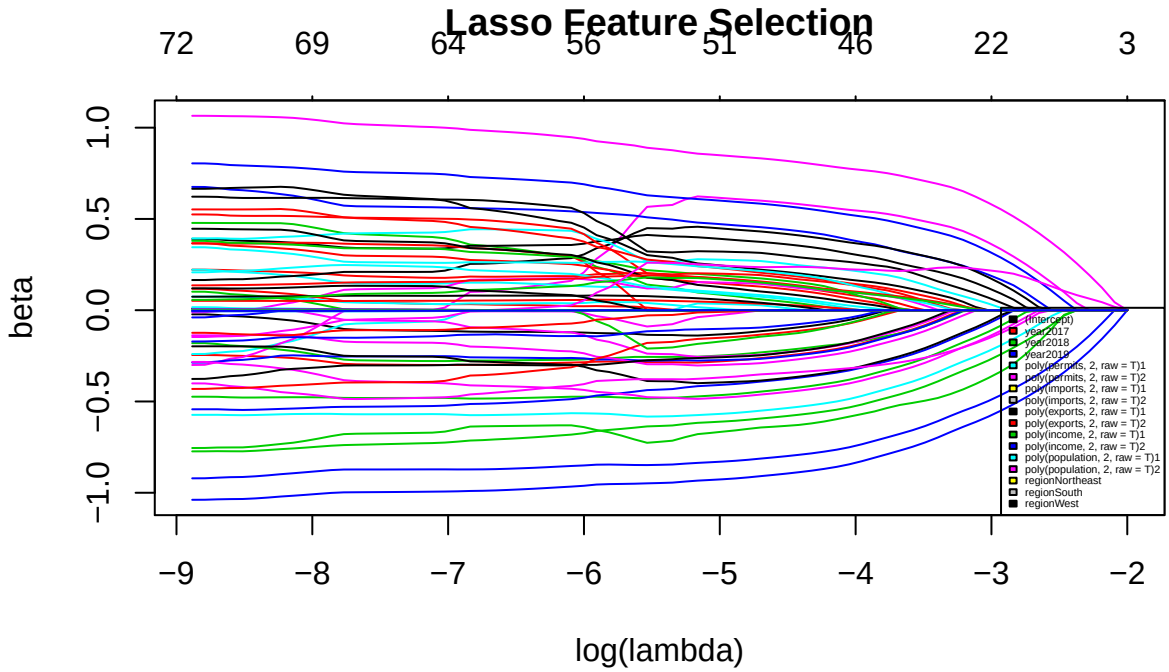
```r
# predictive results
poly1.train <- rmse(predict(polynomial1, train), log(train$overdoseDeaths))
poly1.test <- rmse(predict(polynomial1, test), log(test$overdoseDeaths))
poly2.train <- rmse(predict(polynomial2, train), log(train$overdoseDeaths))
poly2.test <- rmse(predict(polynomial2, test), log(test$overdoseDeaths))

# check model assumptions
plot(polynomial2, which = 3, pch = 19, col = rgb(0, 0, 0, 0.3), sub.caption="")
```



With an F-statistic of 17.359 (with 1881, 2 degrees of freedom) and a corresponding p-value < 0.001, unemployment provides significant predictive ability. Additionally, there seems to be no underlying structure to the residuals, the residuals appear to have reasonably constant variance, and are reasonably normally distributed: the assumptions of our models (linearity, homoscedasticity of residuals, and normality) appear to be satisfied.

In order to be thorough, we examine the importance of unemployment in two additional ways: lasso regression, and random forrest regression. We begin with lasso regression. Due to l1 penalty of lasso regression, less important variables quickly converge to zero as the regularization parameter increases. Below, we fit a lasso model (with the same dedign matrix as that of the full quadratic kitchen sink model) with a series of possible regularization parameters. We then consider the order in which the variable coefficients shrink to zero.

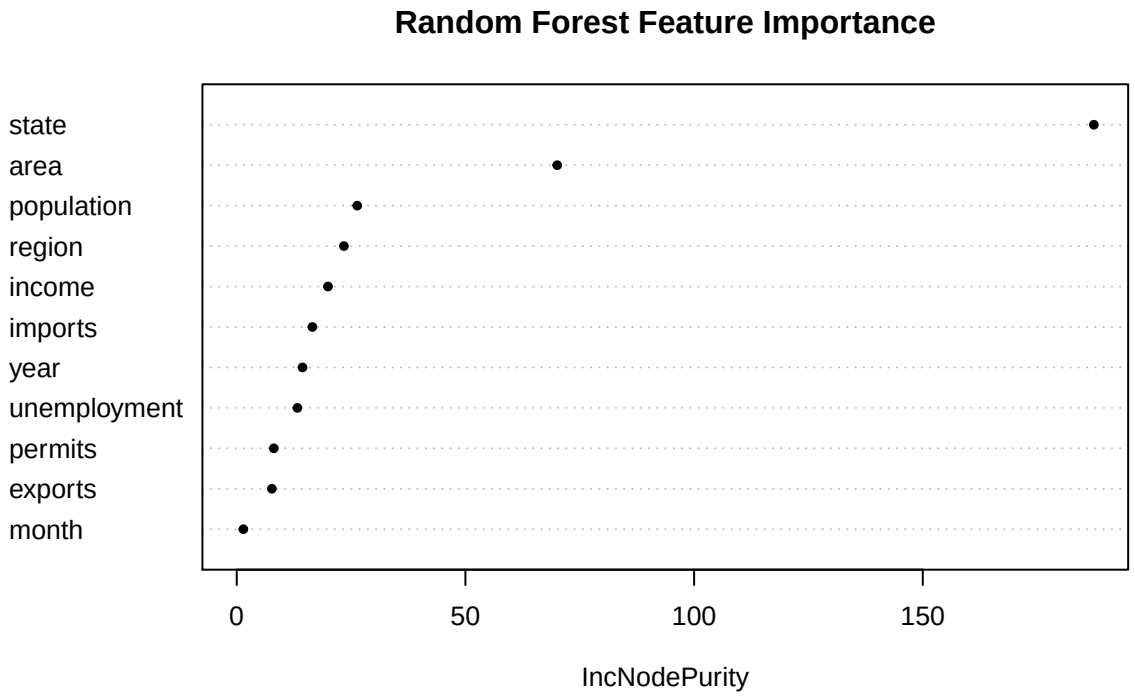
```r
# design matrix and model
X <- model.matrix(formula(polynomial2), data = train)
y <- log(train$overdoseDeaths)
lasso.cv <- cv.glmnet(X, y, alpha = 1)

# feature importance plot
cols <- c(1,64:79)
plot(lasso.cv$glmnet.fit, "lambda", main = "Lasso Feature Selection",
     xlab = "log(lambda)", ylab = "beta")
legend("bottomright", colnames(X)[c(1,64:79)], col = seq_len(length(cols)),
       fill = seq_len(length(cols)), cex = 0.3)
```




```
# predictive results
lasso.train <- sqrt(min(lasso.cv$cvm))
testX <- model.matrix(formula(polynomial2), data = test)
testY <- predict(lasso.cv, newx = testX)
lasso.test <- rmse(testY, log(test$overdoseDeaths))
```

Lasso feature selection agrees with the results of the ESS F-test. Unemployment (the quadratic effect in this case) is one of the last coefficients to shrink to zero along with population. We further test this conclusion with a random forest model. Below, we fit a random forest model with all main effects and then examine it's relative feature importance.

```
# random forest
rf1 <- randomForest(log(overdoseDeaths) ~ state + month + year +
                    permits + imports + exports + income +
                    population + region + area + unemployment,
                    data = train, mtry = 3, ntree = 500)

rf1.train <- rmse(rf1$predicted, log(train$overdoseDeaths))
rf1.test <- rmse(predict(rf1, test), log(test$overdoseDeaths))

# feature importance
varImpPlot(rf1, main = "Random Forest Feature Importance", pch = 20,
           cex = 0.8)
```



Unemployment is relatively not important in the random forest model. It appears as though state and area (which is correlated with state) are the most important predictors. In order to handle the grouped nature of time, state, area, and so forth, we fit a mixed effects model.

Below, we fit a three-layered mixed-effects model. The first two layers are state and year respectively. Theoretically, we would prefer to fit a four-layered model which includes month below year. This is impossible with our dataset ($n = 2448$). With 51 states, 5 years, and 12 months, we would need a minimum of $51 \times 5 \times 12 = 3060$ samples in order to fit such am model. We include a random intercept for states and years. The only random effect we include in the model is unemployment. These are purposeful design choices considering the limited size of our dataset. More complex models would not be reasonably fit.

```
# mixed effects model
lmer1 <- lmer(log(overdoseDeaths) ~ month +
                permits + imports + exports + income +
                population + region + area + unemployment +
                (1 + unemployment | state/year), data = train)

# record results
lmer1.train <- rmse(predict(lmer1, newdata = train), log(train$overdoseDeaths))
lmer1.test <- rmse(predict(lmer1, newdata = test), log(test$overdoseDeaths))

# variable significance
anova(lmer1)
```

```
## Analysis of Variance Table
##              Df   Sum Sq   Mean Sq F value
## month        11 0.305188 0.0277444 20.0454
## permits       1 0.003387 0.0033866  2.4469
## imports       1 0.002240 0.0022399  1.6183
## exports       1 0.000181 0.0001809  0.1307
## income        1 0.012506 0.0125057  9.0354
## population    1 0.002036 0.0020364  1.4713
## region        3 0.009944 0.0033146  2.3948
```

```
## area          1 0.003675 0.0036749  2.6551
## unemployment  1 0.000444 0.0004445  0.3211
```

```
# unemployment coefficient
fixef(lmer1)["unemployment"]
```

```
## unemployment
##  -0.01021003
```

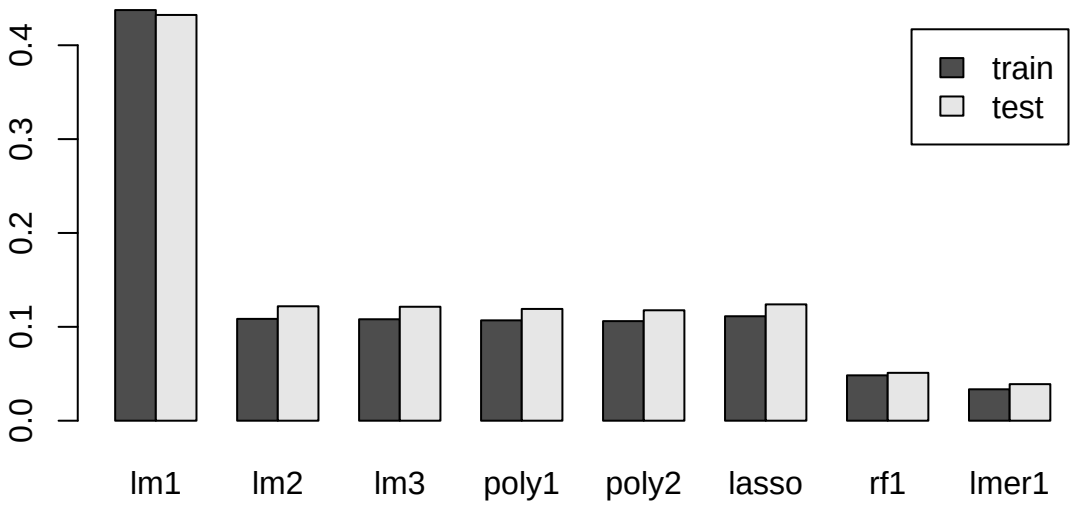
We are not able to calculate p-values for the anova table (the statistics from a mixed model are not F-distributed), however, we can reasonably say that month is the most significant predictor followed by per capita income and region. Unemployment is relatively significant, however, we can show its significance using a p-value. In this model, unemployment has a negative coefficent. This contradicts our simple model and our hypothesis.

Results

```
rmse.overdose <- data.frame(
  train = c(lm1.train, lm2.train, lm3.train, poly1.train, poly2.train,
            lasso.train, rf1.train, lmer1.train),
  test = c(lm1.test, lm2.test, lm3.test, poly1.test, poly2.test, lasso.test,
           rf1.test, lmer1.test),
  row.names = c("lm1", "lm2", "lm3", "poly1", "poly2", "lasso", "rf1", "lmer1")
)
rmse.overdose
```

```
##            train       test
## lm1   0.43747067 0.43228048
## lm2   0.10848281 0.12190404
## lm3   0.10806934 0.12138662
## poly1 0.10689672 0.11906672
## poly2 0.10609875 0.11763930
## lasso 0.11127345 0.12392005
## rf1   0.04833552 0.05100842
## lmer1 0.03348137 0.03896440
```

```
barplot(t(rmse.overdose), beside = T, legend.text = T)
```



Conclusions and Decisions

The Significance of Unemployment:

Unemployment is a significant preditor of overdose death rates. Our simple linear model agrees with this, however, our goal is to show that this relationship holds when other predictors are considered. We thus fit two kitchen sink linear models: one which includes unemployment, and one which does not. An ESS F-test showed that unemplyment provides significant predictive power. We corroborated this result with the same analysis on two quadratic kitchen sink models (again, one included unemployment while the other did not). The resulting ESS F-test agreed with that of the linear models.

We further explored this result with two additional models: a lasso regression model and a random forest regression model. The lasso model showed that unemployment is one of the later predictors to shrink to zero. This coroborates its significance. The random forest model, however, gave much more importance to state, and other non-monthly variables. We can control for these varaibles with a mixed effects model.

Our mixed effects model controled for state and year grouping (with random intercepts) as well as included a random slope for unemployment. The anova table of the resulting model (though limited due to data contraints) showed that unemployment is likely significant. The coefficient for unemployment in this model is negative. We are therefore uncertain as to the association between unemployment and overdose deaths.

Prediction:

The least predictive model is the simple linear model. The rest of the linear models (including the quadratic and lasso-regularized models) performed very similarlly on the test and train sets. This indicates that overfitting is not a serious concern for any of our models. The random forest model and the mixed effects model performed the best reducing RMSE by more than half that of the multiples linear models.

Direction of Future Research

Sample Size:

Month is the most important fixed effect in the mixed model. The predictive ability of the model may increase if month is included as a grouping variable. In our analysis, this was not possible due to limited sample size. In the future, it will be possible to fit such a model.

Predictors:

The insignificance of many of our predictors in the mixed model indicates that we are likely missing important variables for predicting overdose death rates. Some potential predictors are: average temperature, crime rate, high school graduation rate, and literacy rate among others. As of now, these predictors are either not readily available online, not current, or not available in a reasonably frequent timeserie. As more data comes out, variables such as these may add significant predictive power.

References

Ghertner, R., and L. Groves. 2018. "The Opioid Crisis and Economic Opportunity: Geographic and Economic Trends." *Office of the Assistant Secretary for Planning and Execution* 24 (September).

Hedegaard, A. M., H. M.D. Miniño, and M. Ph.D. Warner. 2018. "Drug Overdose Deaths in the United STates, 1999 - 2017." *Centers for Disease Control and Prevention* 329 (November).