Baseline and Regression Models

12/07/2019

```
# load necessary packages

library(readr)
library(dplyr)
library(GGally)
library(ggplot2)
library(car)
```

```
us <- read_csv("https://raw.githubusercontent.com/kanam12/ieor142finalproject/master/us_suicides_merged
#names(suicides)[9] <- "suicides_rate"

suicides <- us %>%  select(-age, - `country-year`, -country)

set.seed(377)


train.ids = sample(nrow(suicides), 0.70*nrow(suicides))
train = suicides[train.ids,]
test = suicides[-train.ids,]
```

Baseline Model

```
base_mod <- mean(suicides$`suicides/100k pop`)
```

Linear Regression

```
set.seed(377)

exp_mod <- lm(`suicides/100k pop` ~ ., data = train)

summary(exp_mod)
```

```
##
## Call:
## lm(formula = `suicides/100k pop` ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.9676  -3.2243   0.1071   2.9260  21.1764
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -6.561e+02  1.330e+03  -0.493  0.62221
## year                       2.763e-01  7.131e-01   0.387  0.69873
## sexmale                    8.961e+00  9.126e+00   0.982  0.32713
## suicides_no                2.155e-03  3.863e-04   5.579 6.39e-08 ***
```

```
## population                -6.970e-07  8.892e-08  -7.839 1.39e-13 ***
## `HDI for year`             1.605e+02  1.736e+02   0.925  0.35600
## `gdp_for_year ($)`         2.110e-12  2.369e-12   0.891  0.37394
## `gdp_per_capita ($)`      -1.112e-03  8.393e-04  -1.325  0.18633
## generationBoomers          5.670e+00  1.378e+00   4.113 5.33e-05 ***
## generationSilent           3.280e+00  1.241e+00   2.643  0.00875 **
## generationG.I. Generation  1.108e+01  1.713e+00   6.469 5.33e-10 ***
## generationMillenials      -2.833e+00  1.360e+00  -2.083  0.03825 *
## generationGeneration Z    -6.841e+00  2.624e+00  -2.607  0.00969 **
## depression_percentage     -3.811e-01  3.736e+00  -0.102  0.91884
## drug_death_rate            1.284e-01  7.705e-02   1.666  0.09699 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.205 on 245 degrees of freedom
## Multiple R-squared:  0.8063, Adjusted R-squared:  0.7952
## F-statistic: 72.84 on 14 and 245 DF,  p-value: < 2.2e-16
```

```r
vif(exp_mod) #- perfect multicolinearity
```

```
##                             GVIF Df GVIF^(1/(2*Df))
## year                  283.510832  1       16.837780
## sex                   140.526937  1       11.854406
## suicides_no             7.524014  1        2.742994
## population              5.056416  1        2.248648
## `HDI for year`        108.411003  1       10.412060
## `gdp_for_year ($)`    688.789938  1       26.244808
## `gdp_per_capita ($)`  747.303383  1       27.336850
## generation              6.593310  5        1.207565
## depression_percentage 140.933912  1       11.871559
## drug_death_rate         7.158119  1        2.675466
```

```r
#alias(exp_mod)
```

```r
set.seed(377)
lin_mod <- lm(`suicides/100k pop` ~ .-`gdp_per_capita ($)`, data = train)

summary(lin_mod)
```

```
##
## Call:
## lm(formula = `suicides/100k pop` ~ . - `gdp_per_capita ($)`,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4011  -2.8929  -0.0201   3.1459  21.7702
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -8.033e+02  1.327e+03  -0.605  0.54555
## year                       4.072e-01  7.073e-01   0.576  0.56535
## sexmale                    9.896e+00  9.113e+00   1.086  0.27856
## suicides_no                2.207e-03  3.849e-04   5.735 2.85e-08 ***
## population                -7.068e-07  8.875e-08  -7.964 6.18e-14 ***
## `HDI for year`             1.158e+01  1.325e+02   0.087  0.93041
```

```
## `gdp_for_year ($)`        -6.552e-13  1.123e-12  -0.583  0.56026
## generationBoomers           5.617e+00  1.380e+00   4.070 6.33e-05 ***
## generationSilent            3.237e+00  1.242e+00   2.606  0.00973 **
## generationG.I. Generation  1.098e+01  1.714e+00   6.404 7.62e-10 ***
## generationMillenials       -2.814e+00  1.362e+00  -2.067  0.03983 *
## generationGeneration Z     -6.503e+00  2.615e+00  -2.486  0.01357 *
## depression_percentage       9.775e-02  3.724e+00   0.026  0.97908
## drug_death_rate             1.313e-01  7.713e-02   1.703  0.08988 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.215 on 246 degrees of freedom
## Multiple R-squared:  0.8049, Adjusted R-squared:  0.7946
## F-statistic: 78.06 on 13 and 246 DF,  p-value: < 2.2e-16
```

```
vif(lin_mod)
```

```
##                             GVIF Df GVIF^(1/(2*Df))
## year                  278.073514  1       16.675536
## sex                   139.686347  1       11.818898
## suicides_no             7.446130  1        2.728760
## population              5.021576  1        2.240887
## `HDI for year`         62.964063  1        7.934990
## `gdp_for_year ($)`    154.381650  1       12.425041
## generation              6.499695  5        1.205839
## depression_percentage 139.615852  1       11.815915
## drug_death_rate         7.152034  1        2.674329
```

```
set.seed(377)
lin_mod2 <- lm(`suicides/100k pop` ~ .-`gdp_per_capita ($)` - year, data = train)

summary(lin_mod2)
```

```
##
## Call:
## lm(formula = `suicides/100k pop` ~ . - `gdp_per_capita ($)` -
##     year, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1604  -3.1161   0.0812   2.8675  21.6604
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -4.094e+01  8.748e+01  -0.468   0.6402
## sexmale                    8.762e+00  8.885e+00   0.986   0.3250
## suicides_no                2.218e-03  3.840e-04   5.776 2.29e-08 ***
## population                -7.041e-07  8.850e-08  -7.955 6.44e-14 ***
## `HDI for year`             6.657e+01  9.169e+01   0.726   0.4685
## `gdp_for_year ($)`        -7.668e-14  5.014e-13  -0.153   0.8786
## generationBoomers          5.583e+00  1.377e+00   4.055 6.72e-05 ***
## generationSilent           3.218e+00  1.240e+00   2.595   0.0100 *
## generationG.I. Generation  1.102e+01  1.710e+00   6.443 6.07e-10 ***
## generationMillenials      -2.785e+00  1.359e+00  -2.049   0.0415 *
## generationGeneration Z    -6.641e+00  2.601e+00  -2.553   0.0113 *
```

```
## depression_percentage     -3.698e-01  3.630e+00  -0.102   0.9189
## drug_death_rate            1.282e-01  7.683e-02   1.668   0.0966 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.206 on 247 degrees of freedom
## Multiple R-squared:  0.8046, Adjusted R-squared:  0.7951
## F-statistic: 84.77 on 12 and 247 DF,  p-value: < 2.2e-16
```

```r
vif(lin_mod2)
```

```
##                             GVIF Df GVIF^(1/(2*Df))
## sex                   133.155360  1       11.539296
## suicides_no             7.429999  1        2.725803
## population              5.007900  1        2.237834
## `HDI for year`         30.238331  1        5.498939
## `gdp_for_year ($)`     30.843074  1        5.553654
## generation              6.356335  5        1.203153
## depression_percentage 132.977265  1       11.531577
## drug_death_rate         7.115567  1        2.667502
```

```r
set.seed(377)
lin_mod3 <- lm(`suicides/100k pop` ~ .-`gdp_per_capita ($)` - year -sex, data = train)

summary(lin_mod3)
```

```
##
## Call:
## lm(formula = `suicides/100k pop` ~ . - `gdp_per_capita ($)` -
##     year - sex, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1977  -3.1137   0.0782   3.0937  21.5800
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.068e+01  7.009e+01   0.152  0.87905
## suicides_no                2.241e-03  3.832e-04   5.846 1.58e-08 ***
## population                -7.064e-07  8.847e-08  -7.985 5.25e-14 ***
## `HDI for year`             2.963e+01  8.368e+01   0.354  0.72355
## `gdp_for_year ($)`         1.219e-13  4.592e-13   0.265  0.79090
## generationBoomers          5.563e+00  1.377e+00   4.041 7.09e-05 ***
## generationSilent           3.331e+00  1.235e+00   2.697  0.00747 **
## generationG.I. Generation  1.105e+01  1.710e+00   6.462 5.43e-10 ***
## generationMillenials      -2.703e+00  1.356e+00  -1.993  0.04739 *
## generationGeneration Z    -6.535e+00  2.599e+00  -2.515  0.01254 *
## depression_percentage     -3.906e+00  5.635e-01  -6.931 3.60e-11 ***
## drug_death_rate            1.296e-01  7.681e-02   1.687  0.09293 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.206 on 248 degrees of freedom
## Multiple R-squared:  0.8039, Adjusted R-squared:  0.7952
## F-statistic:  92.4 on 11 and 248 DF,  p-value: < 2.2e-16
```

```r
vif(lin_mod3)
```

```
##                            GVIF Df GVIF^(1/(2*Df))
## suicides_no            7.402857  1        2.720819
## population             5.004313  1        2.237032
## `HDI for year`        25.192748  1        5.019238
## `gdp_for_year ($)`    25.868829  1        5.086141
## generation             6.284809  5        1.201792
## depression_percentage  3.205765  1        1.790465
## drug_death_rate        7.113163  1        2.667051
```

```r
# remove varibles that are not significant
set.seed(377)
lin_mod4 <- lm(`suicides/100k pop` ~ .-`gdp_for_year ($)` - `gdp_per_capita ($)` - year - sex, data = t

summary(lin_mod4)
```

```
##
## Call:
## lm(formula = `suicides/100k pop` ~ . - `gdp_for_year ($)` - `gdp_per_capita ($)` -
##     year - sex, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4038  -3.0834   0.1028   3.0491  21.6051
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -6.596e+00  2.597e+01  -0.254  0.79974
## suicides_no                 2.251e-03  3.805e-04   5.915 1.09e-08 ***
## population                 -7.099e-07  8.730e-08  -8.132 2.00e-14 ***
## `HDI for year`              5.045e+01  2.911e+01   1.733  0.08430 .
## generationBoomers           5.542e+00  1.372e+00   4.040 7.12e-05 ***
## generationSilent            3.358e+00  1.228e+00   2.733  0.00672 **
## generationG.I. Generation   1.106e+01  1.706e+00   6.484 4.76e-10 ***
## generationMillenials       -2.651e+00  1.340e+00  -1.979  0.04895 *
## generationGeneration Z     -6.356e+00  2.505e+00  -2.538  0.01177 *
## depression_percentage      -3.878e+00  5.527e-01  -7.016 2.15e-11 ***
## drug_death_rate             1.327e-01  7.576e-02   1.751  0.08110 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.194 on 249 degrees of freedom
## Multiple R-squared:  0.8038, Adjusted R-squared:  0.7959
## F-statistic:   102 on 10 and 249 DF,  p-value: < 2.2e-16
```

```r
lin_mod5 <- lm(`suicides/100k pop` ~ .-`gdp_for_year ($)`-`HDI for year`-`gdp_per_capita ($)`-year -sex
                                        data = train)

summary(lin_mod5)
```

```
##
## Call:
## lm(formula = `suicides/100k pop` ~ . - `gdp_for_year ($)` - `HDI for year` -
##     `gdp_per_capita ($)` - year - sex - `HDI for year`, data = train)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.289  -3.002  -0.117   3.230  20.683
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.815e+01  2.857e+00  13.354  < 2e-16 ***
## suicides_no               2.110e-03  3.733e-04   5.653 4.29e-08 ***
## population               -7.623e-07  8.225e-08  -9.268  < 2e-16 ***
## generationBoomers         5.602e+00  1.377e+00   4.069 6.33e-05 ***
## generationSilent          3.905e+00  1.192e+00   3.276   0.0012 **
## generationG.I. Generation  9.855e+00  1.564e+00   6.302 1.32e-09 ***
## generationMillenials     -1.682e+00  1.222e+00  -1.376   0.1701
## generationGeneration Z   -4.237e+00  2.195e+00  -1.931   0.0547 .
## depression_percentage    -3.834e+00  5.543e-01  -6.917 3.85e-11 ***
## drug_death_rate           2.253e-01  5.393e-02   4.178 4.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.219 on 250 degrees of freedom
## Multiple R-squared:  0.8014, Adjusted R-squared:  0.7943
## F-statistic: 112.1 on 9 and 250 DF,  p-value: < 2.2e-16
```

```r
# OSR-squared of newest seasonal model
base_predictions <- rep(base_mod, nrow(test))

#**confirm if this is correct
base_SSE = sum((train$`suicides/100k pop` - rep(base_mod, nrow(train)))^2)
base_SST = sum((train$`suicides/100k pop` - mean(train$`suicides/100k pop`))^2)
base_R2 = 1 - base_SSE/base_SST

base_SSE = sum((test$`suicides/100k pop` - base_predictions)^2)
base_SST = sum((test$`suicides/100k pop` - mean(train$`suicides/100k pop`))^2)
base_OSR2 = 1 - base_SSE/base_SST

# this builds a vector of predicted values on the test set
lin_predictions <- predict(lin_mod5, newdata = test)

lin_SSE = sum((test$`suicides/100k pop` - lin_predictions)^2)
lin_SST = sum((test$`suicides/100k pop` - mean(train$`suicides/100k pop`))^2)
lin_OSR2 = 1 - lin_SSE/lin_SST

#####----------- need to compare change in OSR2

exp_predictions <- predict(exp_mod, newdata = test)

exp_SSE = sum((test$`suicides/100k pop` - exp_predictions)^2)
exp_SST = sum((test$`suicides/100k pop` - mean(train$`suicides/100k pop`))^2)
exp_OSR2 = 1 - exp_SSE/exp_SST
# # OSR-sqaured of the initial exploratory model
# exp_predictions <- predict(mod_exp, newdata = wrangler_test)
#
```

```r
# exp_SSE = sum((wrangler_test$WranglerSales - exp_predictions)^2)
# exp_SST = sum((wrangler_test$WranglerSales - mean(wrangler_train$WranglerSales))^2)
# exp_OSR2 = 1 - exp_SSE/exp_SST

# compare change in R-squared and OSR-squared between the two models

#**confirm if R^2 for baseline is correct
R2 <- c("base_R2" = base_R2, "exp_OR2" = summary(exp_mod)$r.squared, "lin_R2" = summary(lin_mod5)$r.squa
R2
```

```
##     base_R2     exp_OR2      lin_R2
## -0.00054829  0.80628055  0.80143721
```

```r
OSR2 <- c("base_OSR2" = base_OSR2, "exp_OSR2" = exp_OSR2, "lin_OSR2" = lin_OSR2)
OSR2
```

```
##   base_OSR2    exp_OSR2    lin_OSR2
## 0.003987337 0.741355507 0.735686902
```

CART, RF, Boosting

12/13/2019

```r
library(dplyr) # data manipulation
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(caTools) # splits
library(ggplot2) # plot graph
library(randomForest) # Random Forest
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(rpart)
library(rpart.plot)
library(caret)
```

```
## Loading required package: lattice
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```r
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```r
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
OSR2 <- function(predictions, test, train) {
  SSE <- sum((test - predictions)^2)
  SST <- sum((test - mean(train))^2)
  r2 <- 1 - SSE/SST
  return(r2)
}
```

```
us <- read.csv("us_suicides_merged_no_na.csv")
```

```
suicide_us <- us %>%  select(year, sex, suicides_no, population, suicides.100k.pop, HDI.for.year, gdp_fo
```

```
suicide_us$year <- as.factor(suicide_us$year)
```

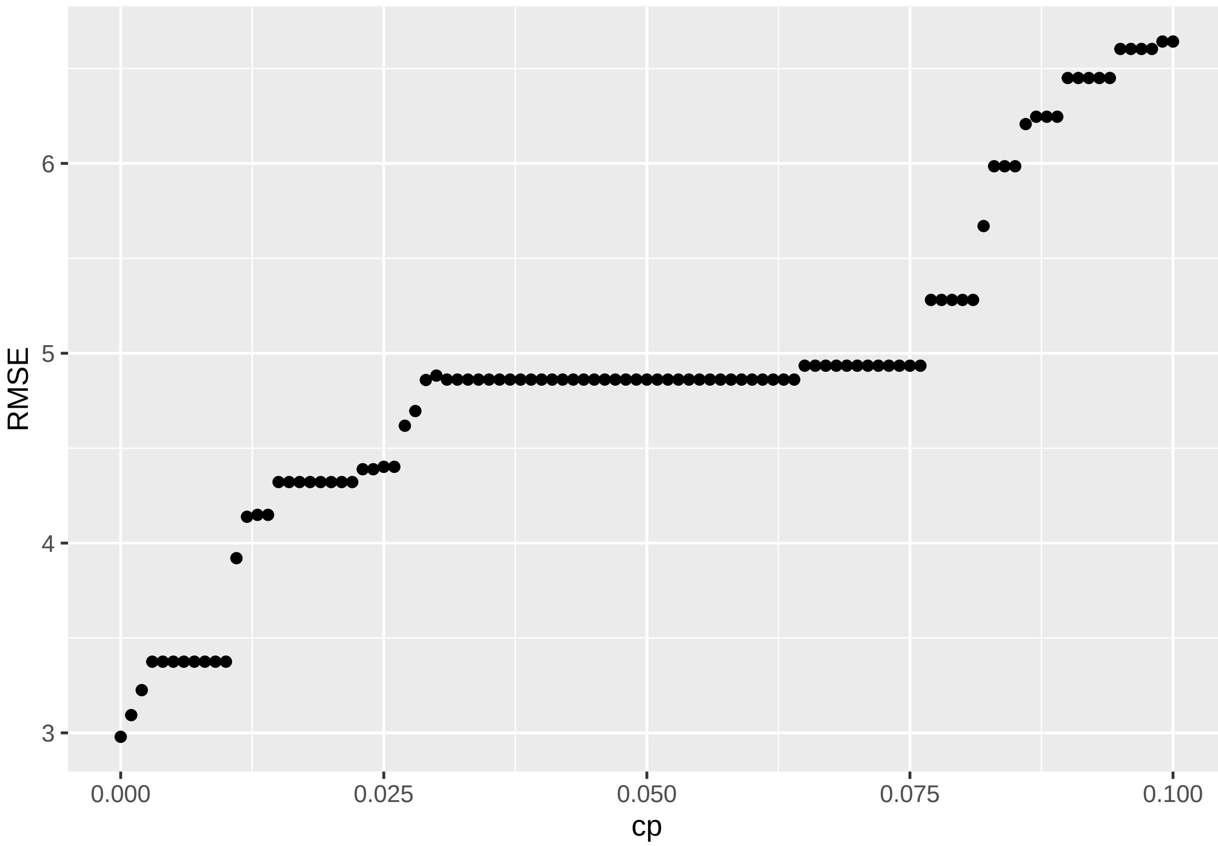
```
# split data for us
set.seed(377)
train.ids_us = sample(nrow(suicide_us), 0.70*nrow(suicide_us))
train_us <- suicide_us[train.ids_us,]
test_us <- suicide_us[-train.ids_us,]
```

CART

```
set.seed(377)

us_train.cart = train(suicides.100k.pop ~ .,
                   data = train_us,
                   method = "rpart",
                   tuneGrid = data.frame(cp=seq(0, 0.1, 0.001)),
                   trControl = trainControl(method="cv", number=10),
                   metric = "RMSE")
us_train.cart$bestTune
```

```
##   cp
## 1  0
```

```
ggplot(us_train.cart$results, aes(x=cp, y=RMSE)) + geom_point()
```



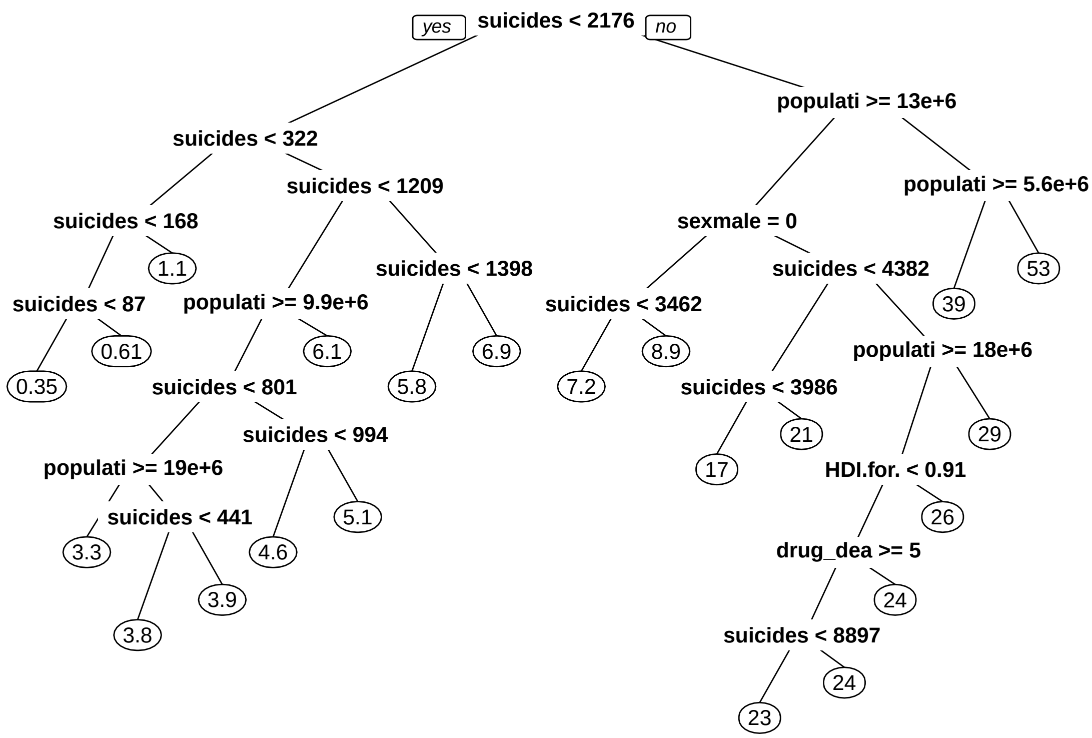

```
us_train.cart$results
```

```
##        cp     RMSE  Rsquared      MAE   RMSESD RsquaredSD     MAESD
## 1   0.000 2.979329 0.9299221 1.199689 3.225845  0.1279826 0.7136592
## 2   0.001 3.093575 0.9277967 1.404896 3.194623  0.1285093 0.6913812
## 3   0.002 3.225177 0.9234075 1.554537 3.217963  0.1315820 0.7476110
## 4   0.003 3.375140 0.9194188 1.738446 3.157827  0.1314394 0.7601279
## 5   0.004 3.375140 0.9194188 1.738446 3.157827  0.1314394 0.7601279
## 6   0.005 3.375140 0.9194188 1.738446 3.157827  0.1314394 0.7601279
## 7   0.006 3.375140 0.9194188 1.738446 3.157827  0.1314394 0.7601279
## 8   0.007 3.375140 0.9194188 1.738446 3.157827  0.1314394 0.7601279
## 9   0.008 3.375140 0.9194188 1.738446 3.157827  0.1314394 0.7601279
## 10  0.009 3.375140 0.9194188 1.738446 3.157827  0.1314394 0.7601279
## 11  0.010 3.375140 0.9194188 1.738446 3.157827  0.1314394 0.7601279
## 12  0.011 3.920498 0.9041719 2.305750 2.903154  0.1267028 0.6481499
## 13  0.012 4.138473 0.8952126 2.487929 2.919139  0.1304562 0.7615112
## 14  0.013 4.148571 0.8954976 2.485649 2.928927  0.1289138 0.7723281
## 15  0.014 4.148571 0.8954976 2.485649 2.928927  0.1289138 0.7723281
## 16  0.015 4.321436 0.8799357 2.518585 3.284802  0.1667361 0.8338564
## 17  0.016 4.321436 0.8799357 2.518585 3.284802  0.1667361 0.8338564
## 18  0.017 4.321436 0.8799357 2.518585 3.284802  0.1667361 0.8338564
## 19  0.018 4.321436 0.8799357 2.518585 3.284802  0.1667361 0.8338564
## 20  0.019 4.321436 0.8799357 2.518585 3.284802  0.1667361 0.8338564
## 21  0.020 4.321436 0.8799357 2.518585 3.284802  0.1667361 0.8338564
## 22  0.021 4.321436 0.8799357 2.518585 3.284802  0.1667361 0.8338564
## 23  0.022 4.321436 0.8799357 2.518585 3.284802  0.1667361 0.8338564
## 24  0.023 4.388922 0.8755937 2.607405 3.422550  0.1734971 1.0382248
```

```
## 25  0.024 4.388922 0.8755937 2.607405 3.422550  0.1734971 1.0382248
## 26  0.025 4.401650 0.8750325 2.635935 3.449401  0.1748987 1.0841685
## 27  0.026 4.401650 0.8750325 2.635935 3.449401  0.1748987 1.0841685
## 28  0.027 4.618113 0.8704746 2.761382 3.361284  0.1724605 1.0469658
## 29  0.028 4.695637 0.8680869 2.780571 3.338454  0.1714471 1.0421989
## 30  0.029 4.859037 0.8660092 2.899967 3.245508  0.1704052 0.9986216
## 31  0.030 4.882301 0.8666749 2.910465 3.225638  0.1708483 0.9861274
## 32  0.031 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 33  0.032 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 34  0.033 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 35  0.034 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 36  0.035 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 37  0.036 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 38  0.037 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 39  0.038 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 40  0.039 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 41  0.040 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 42  0.041 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 43  0.042 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 44  0.043 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 45  0.044 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 46  0.045 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 47  0.046 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 48  0.047 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 49  0.048 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 50  0.049 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 51  0.050 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 52  0.051 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 53  0.052 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 54  0.053 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 55  0.054 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 56  0.055 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 57  0.056 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 58  0.057 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 59  0.058 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 60  0.059 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 61  0.060 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 62  0.061 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 63  0.062 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 64  0.063 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 65  0.064 4.861378 0.8679781 2.899746 3.240825  0.1715778 0.9970946
## 66  0.065 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 67  0.066 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 68  0.067 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 69  0.068 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 70  0.069 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 71  0.070 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 72  0.071 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 73  0.072 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 74  0.073 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 75  0.074 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 76  0.075 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 77  0.076 4.934361 0.8606146 3.059289 3.398389  0.1905633 1.3213671
## 78  0.077 5.280960 0.8462010 3.362222 3.443466  0.1883127 1.5240413
```

```
## 79  0.078 5.280960 0.8462010 3.362222 3.443466  0.1883127 1.5240413
## 80  0.079 5.280960 0.8462010 3.362222 3.443466  0.1883127 1.5240413
## 81  0.080 5.280960 0.8462010 3.362222 3.443466  0.1883127 1.5240413
## 82  0.081 5.280960 0.8462010 3.362222 3.443466  0.1883127 1.5240413
## 83  0.082 5.670020 0.8203833 3.656730 3.311591  0.1868571 1.4931868
## 84  0.083 5.985219 0.8024132 3.897906 3.138200  0.1804592 1.3748995
## 85  0.084 5.985219 0.8024132 3.897906 3.138200  0.1804592 1.3748995
## 86  0.085 5.985219 0.8024132 3.897906 3.138200  0.1804592 1.3748995
## 87  0.086 6.207410 0.7891493 4.068535 2.992776  0.1745661 1.2807708
## 88  0.087 6.245957 0.7861506 4.156334 3.060970  0.1780620 1.3742501
## 89  0.088 6.245957 0.7861506 4.156334 3.060970  0.1780620 1.3742501
## 90  0.089 6.245957 0.7861506 4.156334 3.060970  0.1780620 1.3742501
## 91  0.090 6.450051 0.7771554 4.303759 2.948063  0.1716057 1.2971819
## 92  0.091 6.450051 0.7771554 4.303759 2.948063  0.1716057 1.2971819
## 93  0.092 6.450051 0.7771554 4.303759 2.948063  0.1716057 1.2971819
## 94  0.093 6.450051 0.7771554 4.303759 2.948063  0.1716057 1.2971819
## 95  0.094 6.450051 0.7771554 4.303759 2.948063  0.1716057 1.2971819
## 96  0.095 6.603046 0.7687192 4.419024 2.789900  0.1629593 1.1658689
## 97  0.096 6.603046 0.7687192 4.419024 2.789900  0.1629593 1.1658689
## 98  0.097 6.603046 0.7687192 4.419024 2.789900  0.1629593 1.1658689
## 99  0.098 6.603046 0.7687192 4.419024 2.789900  0.1629593 1.1658689
## 100 0.099 6.642118 0.7712096 4.477140 2.754341  0.1657459 1.0781541
## 101 0.100 6.642118 0.7712096 4.477140 2.754341  0.1657459 1.0781541
```

```r
mod.us_cart <- us_train.cart$finalModel
prp(mod.us_cart)
```




```r
us_test.cart = as.data.frame(model.matrix(suicides.100k.pop ~ . + 0, data=test_us))

predcart_us = predict(mod.us_cart, newdata=us_test.cart)
#predcart_us$results
```

```r
cart.tab.us <- table(test_us$suicides.100k.pop, predcart_us)
cart.tab.us
```

```
##        predcart_us
##         0.35125 0.608571428571429 1.09722222222222 3.26 3.75555555555556
##   0.28        1                 0                0    0                0
##   0.31        1                 0                0    0                0
##   0.37        1                 0                0    0                0
##   0.39        1                 0                0    0                0
##   0.41        1                 0                0    0                0
##   0.42        1                 0                0    0                0
##   0.43        0                 1                0    0                0
##   0.44        1                 0                0    0                0
##   0.45        0                 1                0    0                0
##   0.74        0                 1                0    0                0
##   0.78        0                 1                0    0                0
##   0.83        0                 0                1    0                0
##   0.88        0                 0                1    0                0
##   0.92        0                 0                1    0                0
##   1.02        0                 0                2    0                0
##   1.14        0                 0                2    0                0
##   1.24        0                 0                1    0                0
##   1.3         0                 0                1    0                0
##   1.33        0                 0                1    0                0
##   3.03        0                 0                0    1                0
##   3.12        0                 0                0    0                0
##   3.32        0                 0                0    0                0
##   3.61        0                 0                0    0                1
##   3.72        0                 0                0    0                0
##   3.77        0                 0                0    0                1
##   3.85        0                 0                0    0                1
##   3.88        0                 0                0    0                0
##   4           0                 0                0    0                0
##   4.08        0                 0                0    0                0
##   4.23        0                 0                0    0                0
##   4.32        0                 0                0    0                0
##   4.36        0                 0                0    0                0
##   4.59        0                 0                0    0                0
##   4.73        0                 0                0    0                0
##   4.77        0                 0                0    0                0
##   4.81        0                 0                0    0                0
##   4.95        0                 0                0    0                0
##   5.06        0                 0                0    0                0
##   5.07        0                 0                0    0                0
##   5.17        0                 0                0    0                0
##   5.38        0                 0                0    0                0
##   5.47        0                 0                0    0                0
##   5.61        0                 0                0    0                0
##   5.77        0                 0                0    0                0
##   6.02        0                 0                0    0                0
##   6.17        0                 0                0    0                0
##   6.21        0                 0                0    0                0
##   6.29        0                 0                0    0                0
##   6.4         0                 0                0    0                0
```

```
##   6.76         0                 0               0    0                0
##   6.8          0                 0               0    0                0
##   6.91         0                 0               0    0                0
##   7.06         0                 0               0    0                0
##   7.11         0                 0               0    0                0
##   7.3          0                 0               0    0                0
##   7.38         0                 0               0    0                0
##   7.48         0                 0               0    0                0
##   7.83         0                 0               0    0                0
##   7.9          0                 0               0    0                0
##   8.99         0                 0               0    0                0
##   16.09        0                 0               0    0                0
##   16.15        0                 0               0    0                0
##   16.16        0                 0               0    0                0
##   16.64        0                 0               0    0                0
##   17.2         0                 0               0    0                0
##   17.5         0                 0               0    0                0
##   18.01        0                 0               0    0                0
##   19.57        0                 0               0    0                0
##   20           0                 0               0    0                0
##   20.98        0                 0               0    0                0
##   21.26        0                 0               0    0                0
##   21.89        0                 0               0    0                0
##   21.92        0                 0               0    0                0
##   22.24        0                 0               0    0                0
##   22.25        0                 0               0    0                0
##   22.41        0                 0               0    0                0
##   22.61        0                 0               0    0                0
##   22.62        0                 0               0    0                0
##   23.3         0                 0               0    0                0
##   23.45        0                 0               0    0                0
##   23.57        0                 0               0    0                0
##   23.88        0                 0               0    0                0
##   24           0                 0               0    0                0
##   24.01        0                 0               0    0                0
##   24.03        0                 0               0    0                0
##   24.12        0                 0               0    0                0
##   24.62        0                 0               0    0                0
##   24.76        0                 0               0    0                0
##   24.78        0                 0               0    0                0
##   24.86        0                 0               0    0                0
##   25.02        0                 0               0    0                0
##   25.06        0                 0               0    0                0
##   25.48        0                 0               0    0                0
##   25.52        0                 0               0    0                0
##   25.61        0                 0               0    0                0
##   25.62        0                 0               0    0                0
##   26.34        0                 0               0    0                0
##   26.41        0                 0               0    0                0
##   26.52        0                 0               0    0                0
##   26.71        0                 0               0    0                0
##   27.05        0                 0               0    0                0
##   27.93        0                 0               0    0                0
##   28.11        0                 0               0    0                0
```

```
##   36.53      0                0                0    0               0
##   37.11      0                0                0    0               0
##   45.15      0                0                0    0               0
##   52.33      0                0                0    0               0
##   57.85      0                0                0    0               0
##        predcart_us
##         3.93727272727273 4.59416666666667 5.132 5.78153846153846 6.07375
##   0.28                 0                0     0                0       0
##   0.31                 0                0     0                0       0
##   0.37                 0                0     0                0       0
##   0.39                 0                0     0                0       0
##   0.41                 0                0     0                0       0
##   0.42                 0                0     0                0       0
##   0.43                 0                0     0                0       0
##   0.44                 0                0     0                0       0
##   0.45                 0                0     0                0       0
##   0.74                 0                0     0                0       0
##   0.78                 0                0     0                0       0
##   0.83                 0                0     0                0       0
##   0.88                 0                0     0                0       0
##   0.92                 0                0     0                0       0
##   1.02                 0                0     0                0       0
##   1.14                 0                0     0                0       0
##   1.24                 0                0     0                0       0
##   1.3                  0                0     0                0       0
##   1.33                 0                0     0                0       0
##   3.03                 0                0     0                0       0
##   3.12                 1                0     0                0       0
##   3.32                 1                0     0                0       0
##   3.61                 0                0     0                0       0
##   3.72                 1                0     0                0       0
##   3.77                 0                0     0                0       0
##   3.85                 0                0     0                0       0
##   3.88                 0                1     0                0       0
##   4                    0                1     0                0       0
##   4.08                 1                0     0                0       0
##   4.23                 1                0     0                0       0
##   4.32                 0                1     0                0       0
##   4.36                 0                1     0                0       0
##   4.59                 1                0     0                0       0
##   4.73                 0                1     0                0       0
##   4.77                 0                1     1                0       0
##   4.81                 0                0     0                0       1
##   4.95                 0                0     1                0       0
##   5.06                 0                0     1                0       0
##   5.07                 0                0     0                0       1
##   5.17                 1                0     0                0       0
##   5.38                 0                0     2                0       0
##   5.47                 0                0     0                0       1
##   5.61                 0                0     0                1       0
##   5.77                 0                0     0                0       0
##   6.02                 0                0     0                0       0
##   6.17                 0                0     0                0       1
##   6.21                 0                0     0                1       0
```

```
##   6.29                 0                0     0                1       0
##   6.4                  0                0     0                0       1
##   6.76                 0                0     0                0       0
##   6.8                  0                0     0                0       0
##   6.91                 0                0     0                0       0
##   7.06                 0                0     0                0       0
##   7.11                 0                0     0                0       0
##   7.3                  0                0     0                0       0
##   7.38                 0                0     0                0       0
##   7.48                 0                0     0                0       0
##   7.83                 0                0     0                0       0
##   7.9                  0                0     0                0       0
##   8.99                 0                0     0                0       0
##   16.09                0                0     0                0       0
##   16.15                0                0     0                0       0
##   16.16                0                0     0                0       0
##   16.64                0                0     0                0       0
##   17.2                 0                0     0                0       0
##   17.5                 0                0     0                0       0
##   18.01                0                0     0                0       0
##   19.57                0                0     0                0       0
##   20                   0                0     0                0       0
##   20.98                0                0     0                0       0
##   21.26                0                0     0                0       0
##   21.89                0                0     0                0       0
##   21.92                0                0     0                0       0
##   22.24                0                0     0                0       0
##   22.25                0                0     0                0       0
##   22.41                0                0     0                0       0
##   22.61                0                0     0                0       0
##   22.62                0                0     0                0       0
##   23.3                 0                0     0                0       0
##   23.45                0                0     0                0       0
##   23.57                0                0     0                0       0
##   23.88                0                0     0                0       0
##   24                   0                0     0                0       0
##   24.01                0                0     0                0       0
##   24.03                0                0     0                0       0
##   24.12                0                0     0                0       0
##   24.62                0                0     0                0       0
##   24.76                0                0     0                0       0
##   24.78                0                0     0                0       0
##   24.86                0                0     0                0       0
##   25.02                0                0     0                0       0
##   25.06                0                0     0                0       0
##   25.48                0                0     0                0       0
##   25.52                0                0     0                0       0
##   25.61                0                0     0                0       0
##   25.62                0                0     0                0       0
##   26.34                0                0     0                0       0
##   26.41                0                0     0                0       0
##   26.52                0                0     0                0       0
##   26.71                0                0     0                0       0
##   27.05                0                0     0                0       0
```

```
##   27.93                0                0     0                0       0
##   28.11                0                0     0                0       0
##   36.53                0                0     0                0       0
##   37.11                0                0     0                0       0
##   45.15                0                0     0                0       0
##   52.33                0                0     0                0       0
##   57.85                0                0     0                0       0
##        predcart_us
##         6.88636363636364 7.1625 8.865 16.9827272727273 21.1638888888889
##   0.28                 0      0     0                0                0
##   0.31                 0      0     0                0                0
##   0.37                 0      0     0                0                0
##   0.39                 0      0     0                0                0
##   0.41                 0      0     0                0                0
##   0.42                 0      0     0                0                0
##   0.43                 0      0     0                0                0
##   0.44                 0      0     0                0                0
##   0.45                 0      0     0                0                0
##   0.74                 0      0     0                0                0
##   0.78                 0      0     0                0                0
##   0.83                 0      0     0                0                0
##   0.88                 0      0     0                0                0
##   0.92                 0      0     0                0                0
##   1.02                 0      0     0                0                0
##   1.14                 0      0     0                0                0
##   1.24                 0      0     0                0                0
##   1.3                  0      0     0                0                0
##   1.33                 0      0     0                0                0
##   3.03                 0      0     0                0                0
##   3.12                 0      0     0                0                0
##   3.32                 0      0     0                0                0
##   3.61                 0      0     0                0                0
##   3.72                 0      0     0                0                0
##   3.77                 0      0     0                0                0
##   3.85                 0      0     0                0                0
##   3.88                 0      0     0                0                0
##   4                    0      0     0                0                0
##   4.08                 0      0     0                0                0
##   4.23                 0      0     0                0                0
##   4.32                 0      0     0                0                0
##   4.36                 0      0     0                0                0
##   4.59                 0      0     0                0                0
##   4.73                 0      0     0                0                0
##   4.77                 0      0     0                0                0
##   4.81                 0      0     0                0                0
##   4.95                 0      0     0                0                0
##   5.06                 0      0     0                0                0
##   5.07                 0      0     0                0                0
##   5.17                 0      0     0                0                0
##   5.38                 0      0     0                0                0
##   5.47                 0      0     0                0                0
##   5.61                 0      0     0                0                0
##   5.77                 1      0     0                0                0
##   6.02                 1      0     0                0                0
```

```
##   6.17                   0      0     0                0                0
##   6.21                   0      0     0                0                0
##   6.29                   0      0     0                0                0
##   6.4                    0      0     0                0                0
##   6.76                   0      1     0                0                0
##   6.8                    0      1     0                0                0
##   6.91                   1      0     0                0                0
##   7.06                   1      0     0                0                0
##   7.11                   0      1     0                0                0
##   7.3                    0      1     0                0                0
##   7.38                   0      1     0                0                0
##   7.48                   0      1     0                0                0
##   7.83                   0      1     0                0                0
##   7.9                    1      0     0                0                0
##   8.99                   0      0     1                0                0
##   16.09                  0      0     0                1                0
##   16.15                  0      0     0                1                0
##   16.16                  0      0     0                1                0
##   16.64                  0      0     0                1                0
##   17.2                   0      0     0                1                0
##   17.5                   0      0     0                1                0
##   18.01                  0      0     0                0                1
##   19.57                  0      0     0                1                0
##   20                     0      0     0                1                0
##   20.98                  0      0     0                0                1
##   21.26                  0      0     0                0                1
##   21.89                  0      0     0                0                1
##   21.92                  0      0     0                0                1
##   22.24                  0      0     0                0                1
##   22.25                  0      0     0                0                0
##   22.41                  0      0     0                0                0
##   22.61                  0      0     0                0                0
##   22.62                  0      0     0                0                0
##   23.3                   0      0     0                0                0
##   23.45                  0      0     0                0                1
##   23.57                  0      0     0                0                0
##   23.88                  0      0     0                0                0
##   24                     0      0     0                0                0
##   24.01                  0      0     0                0                0
##   24.03                  0      0     0                0                0
##   24.12                  0      0     0                0                0
##   24.62                  0      0     0                0                0
##   24.76                  0      0     0                0                0
##   24.78                  0      0     0                0                0
##   24.86                  0      0     0                0                0
##   25.02                  0      0     0                0                0
##   25.06                  0      0     0                0                0
##   25.48                  0      0     0                0                0
##   25.52                  0      0     0                0                0
##   25.61                  0      0     0                0                0
##   25.62                  0      0     0                0                0
##   26.34                  0      0     0                0                0
##   26.41                  0      0     0                0                0
##   26.52                  0      0     0                0                0
```

```
##    26.71                0      0     0                0                0
##    27.05                0      0     0                0                0
##    27.93                0      0     0                0                0
##    28.11                0      0     0                0                0
##    36.53                0      0     0                0                0
##    37.11                0      0     0                0                0
##    45.15                0      0     0                0                0
##    52.33                0      0     0                0                0
##    57.85                0      0     0                0                0
##         predcart_us
##          23.0911764705882 24.0785714285714 24.47 26.1814285714286
##    0.28                 0                0     0                0
##    0.31                 0                0     0                0
##    0.37                 0                0     0                0
##    0.39                 0                0     0                0
##    0.41                 0                0     0                0
##    0.42                 0                0     0                0
##    0.43                 0                0     0                0
##    0.44                 0                0     0                0
##    0.45                 0                0     0                0
##    0.74                 0                0     0                0
##    0.78                 0                0     0                0
##    0.83                 0                0     0                0
##    0.88                 0                0     0                0
##    0.92                 0                0     0                0
##    1.02                 0                0     0                0
##    1.14                 0                0     0                0
##    1.24                 0                0     0                0
##    1.3                  0                0     0                0
##    1.33                 0                0     0                0
##    3.03                 0                0     0                0
##    3.12                 0                0     0                0
##    3.32                 0                0     0                0
##    3.61                 0                0     0                0
##    3.72                 0                0     0                0
##    3.77                 0                0     0                0
##    3.85                 0                0     0                0
##    3.88                 0                0     0                0
##    4                    0                0     0                0
##    4.08                 0                0     0                0
##    4.23                 0                0     0                0
##    4.32                 0                0     0                0
##    4.36                 0                0     0                0
##    4.59                 0                0     0                0
##    4.73                 0                0     0                0
##    4.77                 0                0     0                0
##    4.81                 0                0     0                0
##    4.95                 0                0     0                0
##    5.06                 0                0     0                0
##    5.07                 0                0     0                0
##    5.17                 0                0     0                0
##    5.38                 0                0     0                0
##    5.47                 0                0     0                0
##    5.61                 0                0     0                0
```

```
##    5.77                   0                0     0                0
##    6.02                   0                0     0                0
##    6.17                   0                0     0                0
##    6.21                   0                0     0                0
##    6.29                   0                0     0                0
##    6.4                    0                0     0                0
##    6.76                   0                0     0                0
##    6.8                    0                0     0                0
##    6.91                   0                0     0                0
##    7.06                   0                0     0                0
##    7.11                   0                0     0                0
##    7.3                    0                0     0                0
##    7.38                   0                0     0                0
##    7.48                   0                0     0                0
##    7.83                   0                0     0                0
##    7.9                    0                0     0                0
##    8.99                   0                0     0                0
##    16.09                  0                0     0                0
##    16.15                  0                0     0                0
##    16.16                  0                0     0                0
##    16.64                  0                0     0                0
##    17.2                   0                0     0                0
##    17.5                   0                0     0                0
##    18.01                  0                0     0                0
##    19.57                  0                0     0                0
##    20                     0                0     0                0
##    20.98                  0                0     0                0
##    21.26                  0                0     0                0
##    21.89                  0                0     0                0
##    21.92                  0                0     0                0
##    22.24                  0                0     0                0
##    22.25                  1                0     0                0
##    22.41                  1                0     0                0
##    22.61                  1                0     0                0
##    22.62                  0                1     0                0
##    23.3                   1                0     0                0
##    23.45                  0                0     0                0
##    23.57                  0                1     0                0
##    23.88                  0                1     0                0
##    24                     0                0     0                1
##    24.01                  1                0     0                0
##    24.03                  0                0     1                0
##    24.12                  0                1     0                0
##    24.62                  0                0     0                0
##    24.76                  0                0     1                0
##    24.78                  0                0     1                0
##    24.86                  0                0     0                1
##    25.02                  0                0     1                0
##    25.06                  0                0     0                0
##    25.48                  0                0     1                0
##    25.52                  0                0     0                1
##    25.61                  0                0     1                0
##    25.62                  0                0     1                0
##    26.34                  0                0     0                1
```

```
##   26.41                0                0     0                0
##   26.52                0                0     0                1
##   26.71                0                0     0                0
##   27.05                0                0     0                1
##   27.93                0                0     0                1
##   28.11                0                0     0                1
##   36.53                0                0     0                0
##   37.11                0                0     0                0
##   45.15                0                0     0                0
##   52.33                0                0     0                0
##   57.85                0                0     0                0
##        predcart_us
##         29.0328571428571 38.67125 53.46
##   0.28                 0        0     0
##   0.31                 0        0     0
##   0.37                 0        0     0
##   0.39                 0        0     0
##   0.41                 0        0     0
##   0.42                 0        0     0
##   0.43                 0        0     0
##   0.44                 0        0     0
##   0.45                 0        0     0
##   0.74                 0        0     0
##   0.78                 0        0     0
##   0.83                 0        0     0
##   0.88                 0        0     0
##   0.92                 0        0     0
##   1.02                 0        0     0
##   1.14                 0        0     0
##   1.24                 0        0     0
##   1.3                  0        0     0
##   1.33                 0        0     0
##   3.03                 0        0     0
##   3.12                 0        0     0
##   3.32                 0        0     0
##   3.61                 0        0     0
##   3.72                 0        0     0
##   3.77                 0        0     0
##   3.85                 0        0     0
##   3.88                 0        0     0
##   4                    0        0     0
##   4.08                 0        0     0
##   4.23                 0        0     0
##   4.32                 0        0     0
##   4.36                 0        0     0
##   4.59                 0        0     0
##   4.73                 0        0     0
##   4.77                 0        0     0
##   4.81                 0        0     0
##   4.95                 0        0     0
##   5.06                 0        0     0
##   5.07                 0        0     0
##   5.17                 0        0     0
##   5.38                 0        0     0
```

```
## 5.47      0    0    0
## 5.61      0    0    0
## 5.77      0    0    0
## 6.02      0    0    0
## 6.17      0    0    0
## 6.21      0    0    0
## 6.29      0    0    0
## 6.4       0    0    0
## 6.76      0    0    0
## 6.8       0    0    0
## 6.91      0    0    0
## 7.06      0    0    0
## 7.11      0    0    0
## 7.3       0    0    0
## 7.38      0    0    0
## 7.48      0    0    0
## 7.83      0    0    0
## 7.9       0    0    0
## 8.99      0    0    0
## 16.09     0    0    0
## 16.15     0    0    0
## 16.16     0    0    0
## 16.64     0    0    0
## 17.2      0    0    0
## 17.5      0    0    0
## 18.01     0    0    0
## 19.57     0    0    0
## 20        0    0    0
## 20.98     0    0    0
## 21.26     0    0    0
## 21.89     0    0    0
## 21.92     0    0    0
## 22.24     0    0    0
## 22.25     0    0    0
## 22.41     0    0    0
## 22.61     0    0    0
## 22.62     0    0    0
## 23.3      0    0    0
## 23.45     0    0    0
## 23.57     0    0    0
## 23.88     0    0    0
## 24        0    0    0
## 24.01     0    0    0
## 24.03     0    0    0
## 24.12     0    0    0
## 24.62     1    0    0
## 24.76     0    0    0
## 24.78     0    0    0
## 24.86     0    0    0
## 25.02     0    0    0
## 25.06     1    0    0
## 25.48     0    0    0
## 25.52     0    0    0
## 25.61     0    0    0
```

```
##   25.62                 0        0     0
##   26.34                 0        0     0
##   26.41                 1        0     0
##   26.52                 0        0     0
##   26.71                 1        0     0
##   27.05                 0        0     0
##   27.93                 0        0     0
##   28.11                 0        0     0
##   36.53                 0        1     0
##   37.11                 0        1     0
##   45.15                 0        1     0
##   52.33                 0        0     1
##   57.85                 0        0     1
```

```r
print("CART OSR2:")
```

```
## [1] "CART OSR2:"
```

```r
OSR2(predcart_us, test_us$suicides.100k.pop, train_us$suicides.100k.pop)
```

```
## [1] 0.9883818
```

Random Forest

```r
set.seed(377)

mod.rf.us <- randomForest(suicides.100k.pop ~ ., data = train_us, mtry = 5, nodesize = 5, ntree = 500)
```

```r
pred.rf.us <- predict(mod.rf.us, newdata = test_us) # just to illustrate
pred.rf.us[1:5]
```

```
##         1         7        16        21        26
##  4.859536  1.050132 24.295192  7.247386 20.566757
```

```r
importance(mod.rf.us)
```

```
##                          IncNodePurity
## year                          721.5390
## sex                          6873.8985
## suicides_no                 20017.7699
## population                  12227.8713
## HDI.for.year                  248.8317
## gdp_for_year....              268.8887
## gdp_per_capita....            222.9683
## generation                   2319.3296
## depression_percentage        4389.2658
## drug_death_rate              1101.7067
```

```r
set.seed(377)
train.rf.us <- train(suicides.100k.pop ~ .,
                     data = train_us,
                     method = "rf",
                     tuneGrid = data.frame(mtry=1:5),
                     trControl = trainControl(method="cv", number=5, verboseIter = TRUE),
                     metric = "RMSE")
```

```
## + Fold1: mtry=1
## - Fold1: mtry=1
## + Fold1: mtry=2
## - Fold1: mtry=2
## + Fold1: mtry=3
## - Fold1: mtry=3
## + Fold1: mtry=4
## - Fold1: mtry=4
## + Fold1: mtry=5
## - Fold1: mtry=5
## + Fold2: mtry=1
## - Fold2: mtry=1
## + Fold2: mtry=2
## - Fold2: mtry=2
## + Fold2: mtry=3
## - Fold2: mtry=3
## + Fold2: mtry=4
## - Fold2: mtry=4
## + Fold2: mtry=5
## - Fold2: mtry=5
## + Fold3: mtry=1
## - Fold3: mtry=1
## + Fold3: mtry=2
## - Fold3: mtry=2
## + Fold3: mtry=3
## - Fold3: mtry=3
## + Fold3: mtry=4
## - Fold3: mtry=4
## + Fold3: mtry=5
## - Fold3: mtry=5
## + Fold4: mtry=1
## - Fold4: mtry=1
## + Fold4: mtry=2
## - Fold4: mtry=2
## + Fold4: mtry=3
## - Fold4: mtry=3
## + Fold4: mtry=4
## - Fold4: mtry=4
## + Fold4: mtry=5
## - Fold4: mtry=5
## + Fold5: mtry=1
## - Fold5: mtry=1
## + Fold5: mtry=2
## - Fold5: mtry=2
## + Fold5: mtry=3
## - Fold5: mtry=3
## + Fold5: mtry=4
## - Fold5: mtry=4
## + Fold5: mtry=5
## - Fold5: mtry=5
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 5 on full training set
```

```r
train.rf.us$results
```

```
##   mtry     RMSE  Rsquared      MAE    RMSESD  RsquaredSD     MAESD
## 1    1 9.824508 0.8696212 7.994422 0.9381334 0.040652349 0.5209140
## 2    2 6.502285 0.9248426 4.913252 0.8379313 0.007879139 0.4761299
## 3    3 4.769440 0.9495409 3.396341 0.7533864 0.010192692 0.4333536
## 4    4 3.711369 0.9646292 2.495179 0.6784323 0.007789490 0.3832441
## 5    5 3.069317 0.9735388 2.018823 0.5291298 0.005027661 0.3227183
```

```r
best.rf.us <- train.rf.us$finalModel
```

```r
us.test_rf = as.data.frame(model.matrix(suicides.100k.pop ~ . + 0, data = test_us))
```

```r
pred.best.rf_us <- predict(best.rf.us, newdata = us.test_rf)
pred.best.rf_us[1:5]
```

```
##         1         7        16        21        26
##  6.398268  3.707426 23.008092  7.826827 21.054450
```

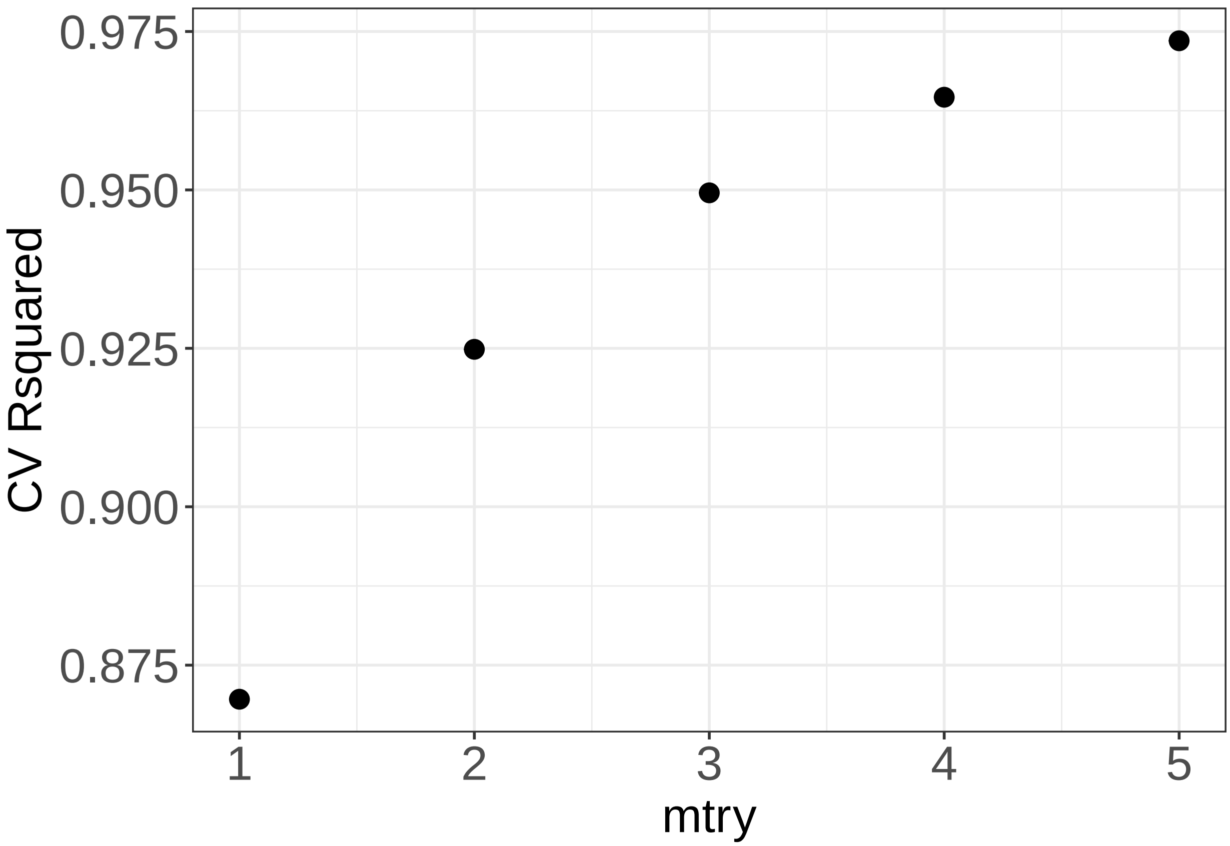
```r
print("Random Forests OSR2:")
```

```
## [1] "Random Forests OSR2:"
```

```r
OSR2(pred.best.rf_us, test_us$suicides.100k.pop, train_us$suicides.100k.pop)
```

```
## [1] 0.9645385
```

```r
ggplot(train.rf.us$results, aes(x = mtry, y = Rsquared)) + geom_point(size = 3) +
  ylab("CV Rsquared") + theme_bw() + theme(axis.title=element_text(size=18), axis.text=element_text(size
```



mtry = 10

```
set.seed(377)
train.rf.us_mtryTen <- train(suicides.100k.pop ~ .,
                             data = train_us,
                             method = "rf",
                             tuneGrid = data.frame(mtry=1:10),
                             trControl = trainControl(method="cv", number=5, verboseIter = TRUE),
                             metric = "RMSE")
```

```
## + Fold1: mtry= 1
## - Fold1: mtry= 1
## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry= 3
## - Fold1: mtry= 3
## + Fold1: mtry= 4
## - Fold1: mtry= 4
## + Fold1: mtry= 5
## - Fold1: mtry= 5
## + Fold1: mtry= 6
## - Fold1: mtry= 6
## + Fold1: mtry= 7
## - Fold1: mtry= 7
## + Fold1: mtry= 8
## - Fold1: mtry= 8
## + Fold1: mtry= 9
## - Fold1: mtry= 9
## + Fold1: mtry=10
## - Fold1: mtry=10
## + Fold2: mtry= 1
## - Fold2: mtry= 1
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry= 3
## - Fold2: mtry= 3
## + Fold2: mtry= 4
## - Fold2: mtry= 4
## + Fold2: mtry= 5
## - Fold2: mtry= 5
## + Fold2: mtry= 6
## - Fold2: mtry= 6
## + Fold2: mtry= 7
## - Fold2: mtry= 7
## + Fold2: mtry= 8
## - Fold2: mtry= 8
## + Fold2: mtry= 9
## - Fold2: mtry= 9
## + Fold2: mtry=10
## - Fold2: mtry=10
## + Fold3: mtry= 1
## - Fold3: mtry= 1
## + Fold3: mtry= 2
## - Fold3: mtry= 2
```

```
## + Fold3: mtry= 3
## - Fold3: mtry= 3
## + Fold3: mtry= 4
## - Fold3: mtry= 4
## + Fold3: mtry= 5
## - Fold3: mtry= 5
## + Fold3: mtry= 6
## - Fold3: mtry= 6
## + Fold3: mtry= 7
## - Fold3: mtry= 7
## + Fold3: mtry= 8
## - Fold3: mtry= 8
## + Fold3: mtry= 9
## - Fold3: mtry= 9
## + Fold3: mtry=10
## - Fold3: mtry=10
## + Fold4: mtry= 1
## - Fold4: mtry= 1
## + Fold4: mtry= 2
## - Fold4: mtry= 2
## + Fold4: mtry= 3
## - Fold4: mtry= 3
## + Fold4: mtry= 4
## - Fold4: mtry= 4
## + Fold4: mtry= 5
## - Fold4: mtry= 5
## + Fold4: mtry= 6
## - Fold4: mtry= 6
## + Fold4: mtry= 7
## - Fold4: mtry= 7
## + Fold4: mtry= 8
## - Fold4: mtry= 8
## + Fold4: mtry= 9
## - Fold4: mtry= 9
## + Fold4: mtry=10
## - Fold4: mtry=10
## + Fold5: mtry= 1
## - Fold5: mtry= 1
## + Fold5: mtry= 2
## - Fold5: mtry= 2
## + Fold5: mtry= 3
## - Fold5: mtry= 3
## + Fold5: mtry= 4
## - Fold5: mtry= 4
## + Fold5: mtry= 5
## - Fold5: mtry= 5
## + Fold5: mtry= 6
## - Fold5: mtry= 6
## + Fold5: mtry= 7
## - Fold5: mtry= 7
## + Fold5: mtry= 8
## - Fold5: mtry= 8
## + Fold5: mtry= 9
## - Fold5: mtry= 9
```

```
## + Fold5: mtry=10
## - Fold5: mtry=10
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 10 on full training set
```

```r
train.rf.us_mtryTen$results
```

```
##    mtry     RMSE  Rsquared      MAE    RMSESD  RsquaredSD     MAESD
## 1     1 9.882207 0.8543650 8.035307 1.0023155 0.036784552 0.6113408
## 2     2 6.620567 0.9187419 4.985777 0.9989036 0.023054109 0.5563487
## 3     3 4.875914 0.9456037 3.456866 0.7017972 0.007218984 0.4125931
## 4     4 3.752569 0.9659937 2.568780 0.6596917 0.006132700 0.3727510
## 5     5 3.000041 0.9743157 1.975137 0.5458457 0.005464792 0.3490734
## 6     6 2.617097 0.9784615 1.669313 0.5002927 0.004488329 0.3125495
## 7     7 2.349818 0.9807843 1.408615 0.4467046 0.006388239 0.2691800
## 8     8 2.211275 0.9820666 1.316461 0.3986787 0.005339521 0.2288858
## 9     9 1.990433 0.9844981 1.167108 0.4789708 0.007932799 0.2326738
## 10   10 1.889257 0.9860799 1.087407 0.3959739 0.005604698 0.2008353
```

```r
best.rf.us_mtryTen <- train.rf.us_mtryTen$finalModel
```

```r
pred.best.rf_us_mtryTen <- predict(best.rf.us_mtryTen, newdata = us.test_rf)
pred.best.rf_us_mtryTen[1:5]
```

```
##         1         7        16        21        26
##  5.047776  1.655800 24.156776  7.160834 20.078536
```

```r
print("Random Forests OSR2:")
```

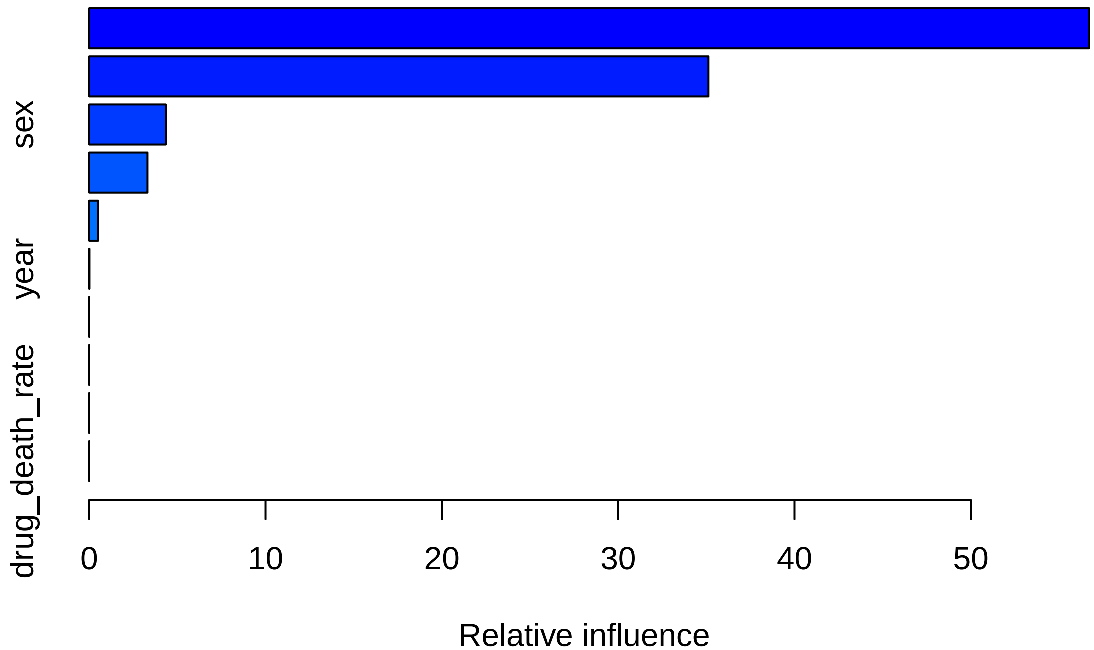
```
## [1] "Random Forests OSR2:"
```

```r
OSR2(pred.best.rf_us_mtryTen, test_us$suicides.100k.pop, train_us$suicides.100k.pop)
```

```
## [1] 0.9928406
```

Boosting

```r
mod.boost <- gbm(suicides.100k.pop ~ .,
                 data = train_us,
                 distribution = "gaussian",
                 n.trees = 1000,
                 shrinkage = 0.001,
                 interaction.depth = 2)
summary(mod.boost)
```




```
##                                               var      rel.inf
## suicides_no                           suicides_no 56.71060064
## population                             population 35.11841374
## sex                                           sex  4.34138880
## depression_percentage depression_percentage  3.30468990
## generation                             generation  0.51025724
## year                                         year  0.01464967
## HDI.for.year                         HDI.for.year  0.00000000
## gdp_for_year....                 gdp_for_year....  0.00000000
## gdp_per_capita....             gdp_per_capita....  0.00000000
## drug_death_rate                   drug_death_rate  0.00000000
```

```
pred.boost <- predict(mod.boost, newdata = test_us, n.trees=1000)
OSR2(pred.boost, test_us$suicides.100k.pop, train_us$suicides.100k.pop)
```

```
## [1] 0.7628486
```

```
## took a while to run -- not super amazing OSR^2
# test_us_mm = as.data.frame(model.matrix(suicides.100k.pop ~ . + 0, data = test_us))
#
# gbmGrid <-  expand.grid(interaction.depth = c(1,2,4,6,8,10),
#                         n.trees = (1:75)*500,
#                         shrinkage = 0.001,
#                         n.minobsinnode = 10)
# fitControl <- trainControl(## 10-fold CV
#                            method = "repeatedcv",
#                            number = 5,
#                            ## repeated ten times
#                            repeats = 5)
# set.seed(377)
# gbmFit2 <- train(suicides.100k.pop ~ ., data = train_us,
#                  method = "gbm",
#                  trControl = fitControl,
#                  verbose = FALSE,
```

```
#                   tuneGrid = gbmGrid)
#
# gbm.best <- gbmFit2$finalModel
# gbm.pred.best.boost <- predict(gbm.best, newdata = test_us_mm, n.trees = 11500)
# OSR2(gbm.pred.best.boost, test_us$suicides.100k.pop, train_us$suicides.100k.pop)
```

```
## same results as above
# tGrid = expand.grid(n.trees = 1000, interaction.depth = 2, shrinkage = 0.001, n.minobsinnode = 10)tGr
#
# set.seed(377)
# train.boost <- train(suicides.100k.pop ~ .,
#                       data = train_us,
#                       method = "gbm",
#                       tuneGrid = tGrid,
#                       trControl = trainControl(method="cv", number=5,
#                                                verboseIter = FALSE),
#                       metric = "RMSE",
#                       distribution = "gaussian",
#                       verbose = FALSE)
# train.boost
# best.boost <- train.boost$finalModel
# pred.best.boost <- predict(best.boost, newdata = test_us_mm, n.trees = 11500) # can use same model ma
#
# ggplot(train.boost$results, aes(x = n.trees, y = Rsquared, colour = as.factor(interaction.depth))) + 
#   ylab("CV Rsquared") + theme_bw() + theme(axis.title=element_text(size=18), axis.text=element_text(s
#   scale_color_discrete(name = "interaction.depth")
```

```
# OSR2(pred.best.boost, test_us$suicides.100k.pop, train_us$suicides.100k.pop)
# #Out-of-sample MAE:
# sum(abs(test_us$suicides.100k.pop - pred.best.boost))/nrow(test_us_mm)
```

timeseries

12/18/2019

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

Time series

```r
OSR2 <- function(predictions, test, train) {
  SSE <- sum((test - predictions)^2)
  SST <- sum((test - mean(train))^2)
  r2 <- 1 - SSE/SST
  return(r2)
}
```

```r
# R^2 with a particular baseline
BaselineR2 <- function(predictions, truth, baseline) {
  SSE <- sum((truth - predictions)^2)
  SST <- sum((truth - baseline)^2)
  r2 <- 1 - SSE/SST
  return(r2)
}
```

```r
# Load data and check it out
us_ts = read.csv("us_suicides_merged_no_na.csv")
str(us_ts)
```

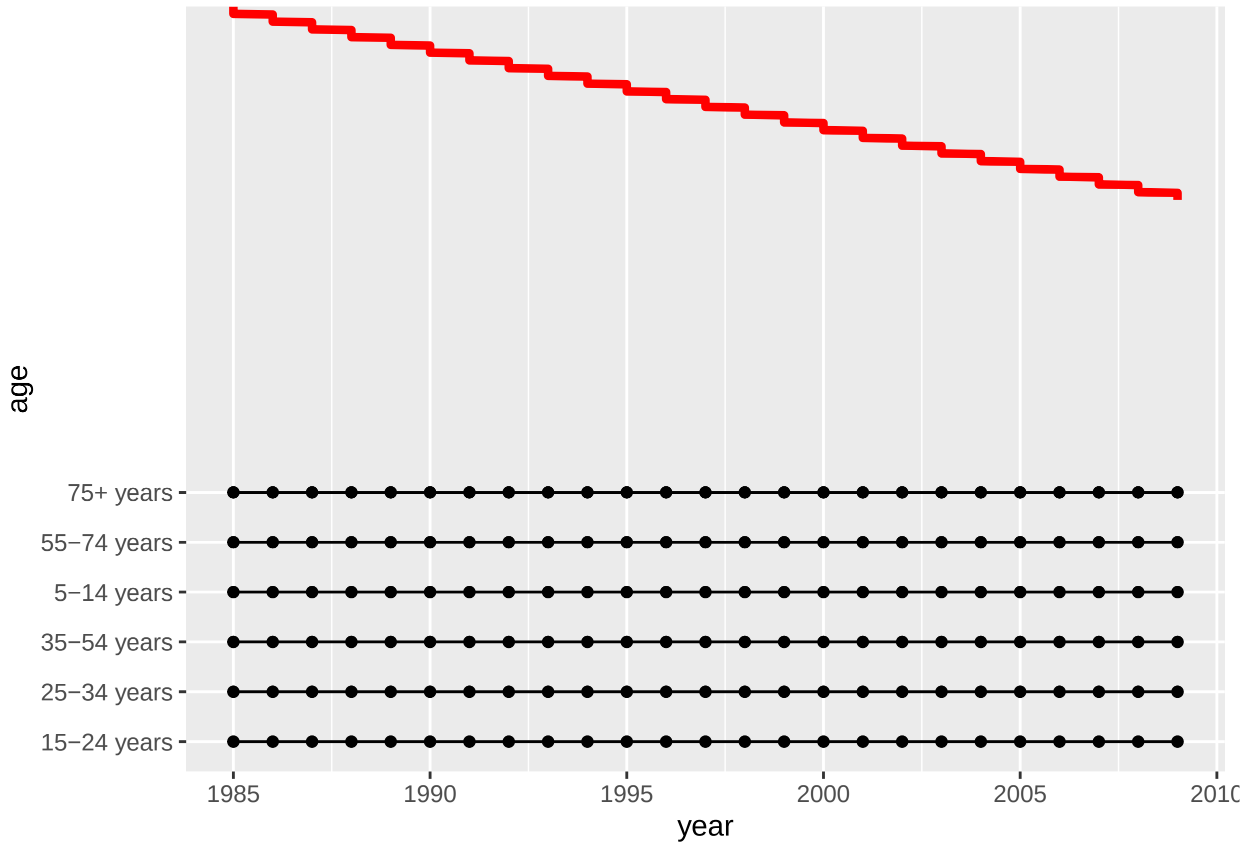
```
## 'data.frame':    372 obs. of  14 variables:
##  $ country              : Factor w/ 1 level "United States": 1 1 1 1 1 1 1 1 1 1 ...
##  $ year                 : int  1985 1985 1985 1985 1985 1985 1985 1985 1985 1985 ...
##  $ sex                  : Factor w/ 2 levels "female","male": 1 2 1 2 1 2 1 2 1 2 ...
##  $ age                  : Factor w/ 6 levels "15-24 years",..: 1 1 2 2 3 3 4 4 5 5 ...
##  $ suicides_no          : int  854 4267 1242 5134 2105 6053 73 205 1568 5302 ...
```

```
##  $ population          : int  19589000 19962000 21041000 20986000 27763000 26589000 16553000 1737000
##  $ suicides.100k.pop   : num  4.36 21.38 5.9 24.46 7.58 ...
##  $ country.year        : Factor w/ 31 levels "United States1985",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ HDI.for.year        : num  0.841 0.841 0.841 0.841 0.841 0.841 0.841 0.841 0.841 0.841 ...
##  $ gdp_for_year....    : num  4.35e+12 4.35e+12 4.35e+12 4.35e+12 4.35e+12 ...
##  $ gdp_per_capita....  : int  19693 19693 19693 19693 19693 19693 19693 19693 19693 19693 ...
##  $ generation          : Factor w/ 6 levels "Boomers","G.I. Generation",..: 3 3 1 1 6 6 3 3 2 2 ...
##  $ depression_percentage: num  6.52 3.52 6.52 3.52 6.52 ...
##  $ drug_death_rate     : num  0 0 0 0 0 ...
```

```
# Use 2013 as testing data
train_ts <- us_ts %>% filter(year < 2010)
test_ts <- us_ts %>% filter(year >= 2010)
```

BUILDING MODELS:

```
# Linear trend model training data -- Make a new column for the time period
# number (1, 2, ...). The dplyr syntax is a little tricky here -- n() is the
# number of rows in salesTrain, and seq_len(n()) returns the vector 1, 2, ...,
# n(). The end result is that we added a new variable called TimePeriod that
# takes values 1, 2, ..., n().
trainLM_ts<- train_ts %>% mutate(TimePeriod = seq_len(n()))
# Build and plot linear trend model
modLM <- lm(suicides.100k.pop~TimePeriod, data=trainLM_ts)
ggplot(trainLM_ts, aes(x=year, y=age)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y=predict(modLM)), col="red", lwd=1.5)
```



Random Walk model training data

```
trainRW_ts <- train_ts %>% mutate(LastYear = c(rep(NA, 12), head(suicides.100k.pop, -12)))
head(trainRW_ts, 15)
```

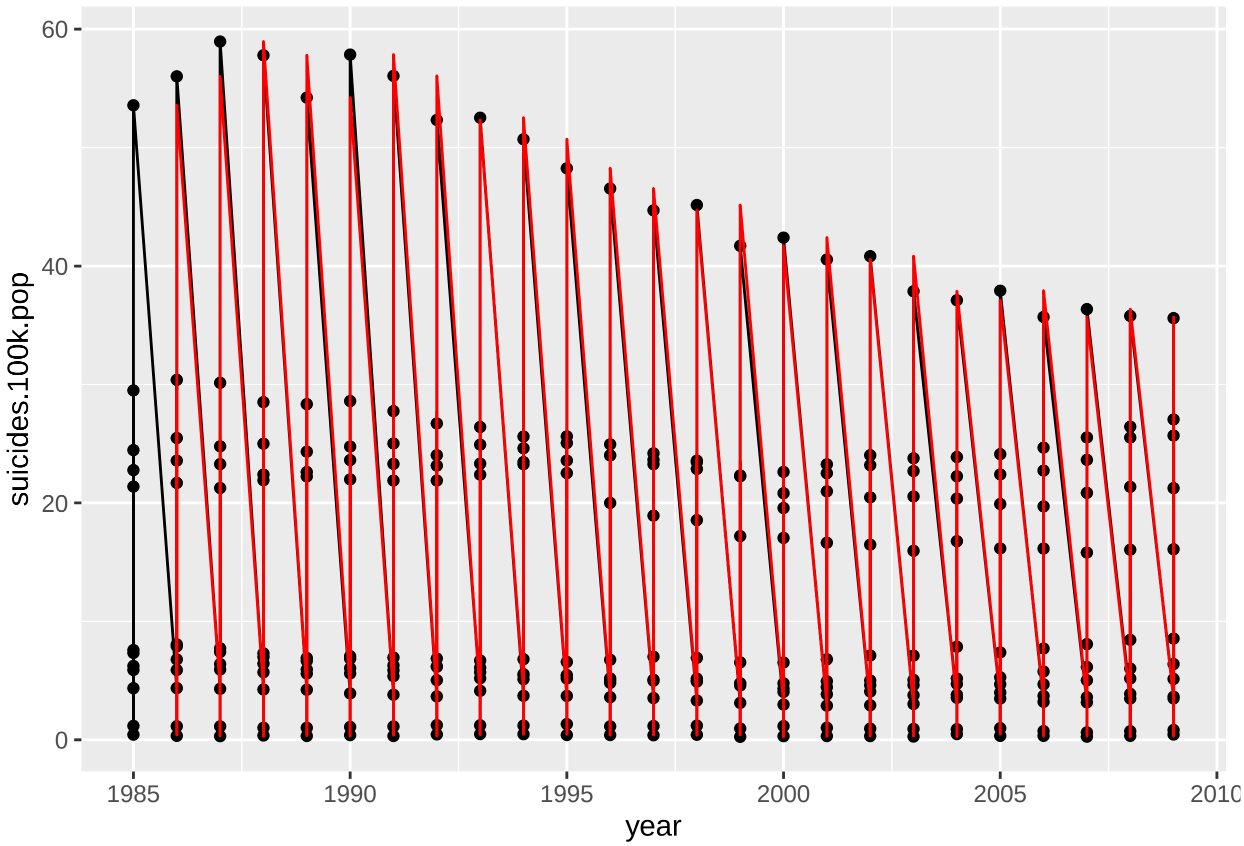
```
##          country year    sex         age suicides_no population
## 1  United States 1985 female 15-24 years         854   19589000
## 2  United States 1985   male 15-24 years        4267   19962000
## 3  United States 1985 female 25-34 years        1242   21041000
## 4  United States 1985   male 25-34 years        5134   20986000
## 5  United States 1985 female 35-54 years        2105   27763000
## 6  United States 1985   male 35-54 years        6053   26589000
## 7  United States 1985 female  5-14 years          73   16553000
## 8  United States 1985   male  5-14 years         205   17370000
## 9  United States 1985 female 55-74 years        1568   21366000
## 10 United States 1985   male 55-74 years        5302   17971000
## 11 United States 1985 female   75+ years         466    7469000
## 12 United States 1985   male   75+ years        2177    4064000
## 13 United States 1986 female 15-24 years         844   19313000
## 14 United States 1986   male 15-24 years        4276   19715000
## 15 United States 1986 female 25-34 years        1261   21391000
##    suicides.100k.pop      country.year HDI.for.year gdp_for_year....
## 1               4.36 United States1985        0.841     4.346734e+12
## 2              21.38 United States1985        0.841     4.346734e+12
## 3               5.90 United States1985        0.841     4.346734e+12
## 4              24.46 United States1985        0.841     4.346734e+12
## 5               7.58 United States1985        0.841     4.346734e+12
```

```
## 6               22.77 United States1985        0.841     4.346734e+12
## 7                0.44 United States1985        0.841     4.346734e+12
## 8                1.18 United States1985        0.841     4.346734e+12
## 9                7.34 United States1985        0.841     4.346734e+12
## 10              29.50 United States1985        0.841     4.346734e+12
## 11               6.24 United States1985        0.841     4.346734e+12
## 12              53.57 United States1985        0.841     4.346734e+12
## 13               4.37 United States1986        0.850     4.590155e+12
## 14              21.69 United States1986        0.850     4.590155e+12
## 15               5.90 United States1986        0.850     4.590155e+12
##    gdp_per_capita....      generation depression_percentage
## 1                19693    Generation X              6.519361
## 2                19693    Generation X              3.520442
## 3                19693         Boomers              6.519361
## 4                19693         Boomers              3.520442
## 5                19693          Silent              6.519361
## 6                19693          Silent              3.520442
## 7                19693    Generation X              6.519361
## 8                19693    Generation X              3.520442
## 9                19693 G.I. Generation              6.519361
## 10               19693 G.I. Generation              3.520442
## 11               19693 G.I. Generation              6.519361
## 12               19693 G.I. Generation              3.520442
## 13               20588    Generation X              6.274631
## 14               20588    Generation X              3.520368
## 15               20588         Boomers              6.274631
##    drug_death_rate LastYear
## 1       0.00000000       NA
## 2       0.00000000       NA
## 3       0.00000000       NA
## 4       0.00000000       NA
## 5       0.00000000       NA
## 6      10.69852941       NA
## 7       0.20000000       NA
## 8       0.20000000       NA
## 9       0.00000000       NA
## 10      0.00000000       NA
## 11      7.46761333       NA
## 12      7.46761333       NA
## 13      0.00000000     4.36
## 14      0.03970588    21.38
## 15      0.00000000     5.90
```

```r
#random walk aka moving average
```

```r
# Plot with an additional red line for our predictions as before
ggplot(trainRW_ts, aes(x=year, y=suicides.100k.pop)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y=LastYear), col="red")
```

```
## Warning: Removed 12 rows containing missing values (geom_path).
```




```
# Proportion of percentages for which difference is more than 1.
table(abs(trainRW_ts$suicides.100k.pop-trainRW_ts$LastYear) >= 1)
```

```
##
## FALSE  TRUE
##   255    33
```

```
# Compute training set R^2
# Note that we need to remove the first observation since there is no
# prediction. This is achieved using tail(.., -1) which says to take all but
# the first observation.
BaselineR2(tail(trainRW_ts$LastYear, -12),
           tail(trainRW_ts$suicides.100k.pop, -12),
           mean(trainRW_ts$suicides.100k.pop))
```

```
## [1] 0.9965203
```

AR model

```
# We need to add sales yesterday and sales two days ago for the two term AR model
# head(.., -2) says take all but the last two
trainAR_ts <- train_ts %>%
  mutate(LastYear=c(rep(NA, 12), head(suicides.100k.pop, -12))) %>%
  mutate(TwoYearsAgo = c(rep(NA, 24), head(suicides.100k.pop, -24)))
# Do the regression with one lag term
mod2a <- lm(suicides.100k.pop~LastYear, data=trainAR_ts)
summary(mod2a)
```

```
##
```

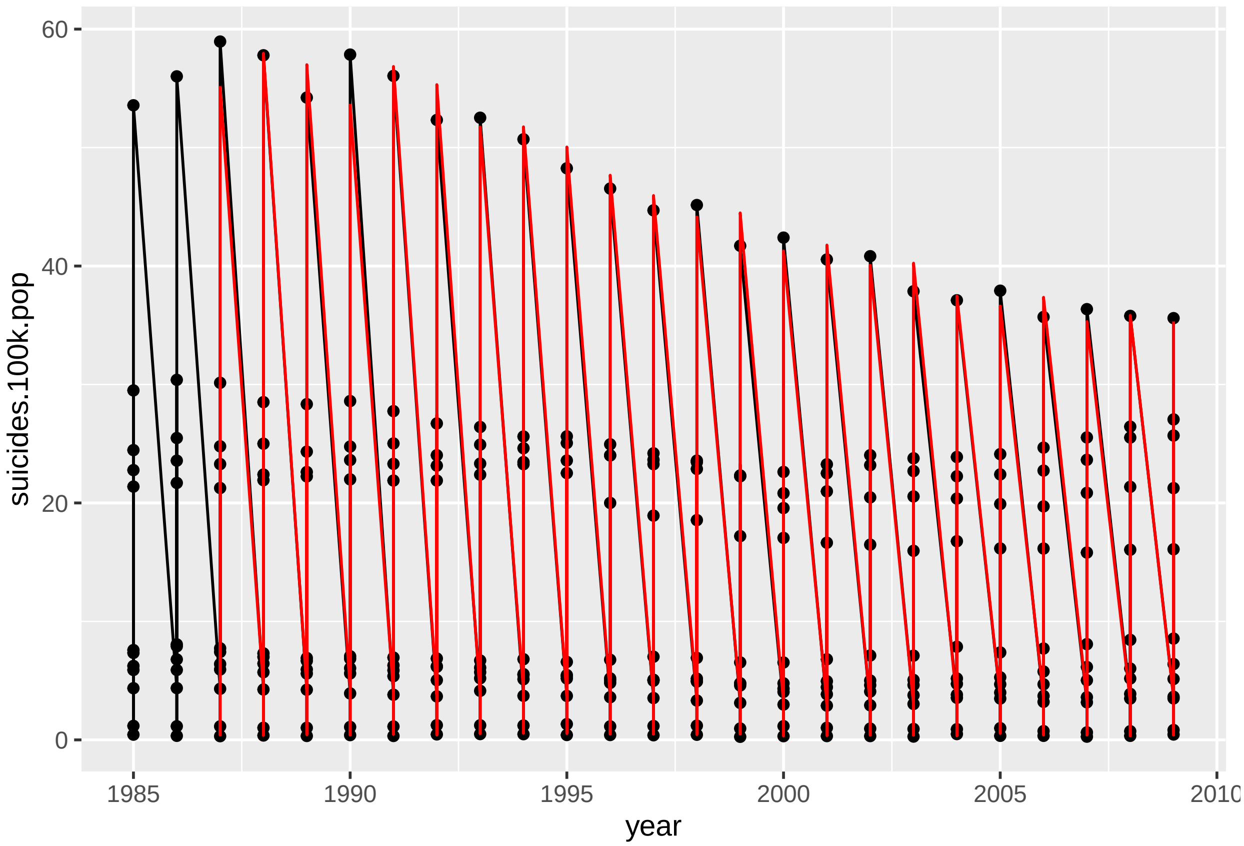
```
## Call:
## lm(formula = suicides.100k.pop ~ LastYear, data = trainAR_ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0903 -0.2250 -0.0333  0.2597  4.2370
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06576    0.06478   1.015    0.311
## LastYear     0.98759    0.00334 295.699   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7699 on 286 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.9967, Adjusted R-squared:  0.9967
## F-statistic: 8.744e+04 on 1 and 286 DF,  p-value: < 2.2e-16
```

```
# 2-term autoregressive model
mod2b <- lm(suicides.100k.pop~LastYear+TwoYearsAgo, data=trainAR_ts)
summary(mod2b)
```

```
##
## Call:
## lm(formula = suicides.100k.pop ~ LastYear + TwoYearsAgo, data = trainAR_ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9838 -0.2199 -0.0377  0.2289  4.2588
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.07916    0.06455   1.226    0.221
## LastYear     0.93990    0.05792  16.229   <2e-16 ***
## TwoYearsAgo  0.04414    0.05727   0.771    0.442
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.751 on 273 degrees of freedom
##   (24 observations deleted due to missingness)
## Multiple R-squared:  0.9969, Adjusted R-squared:  0.9968
## F-statistic: 4.333e+04 on 2 and 273 DF,  p-value: < 2.2e-16
```

```
# Plot with an additional red line for our predictions as before
ggplot(trainAR_ts, aes(x=year, y=suicides.100k.pop)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y=predict(mod2b, newdata=trainAR_ts)), col="red")
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```



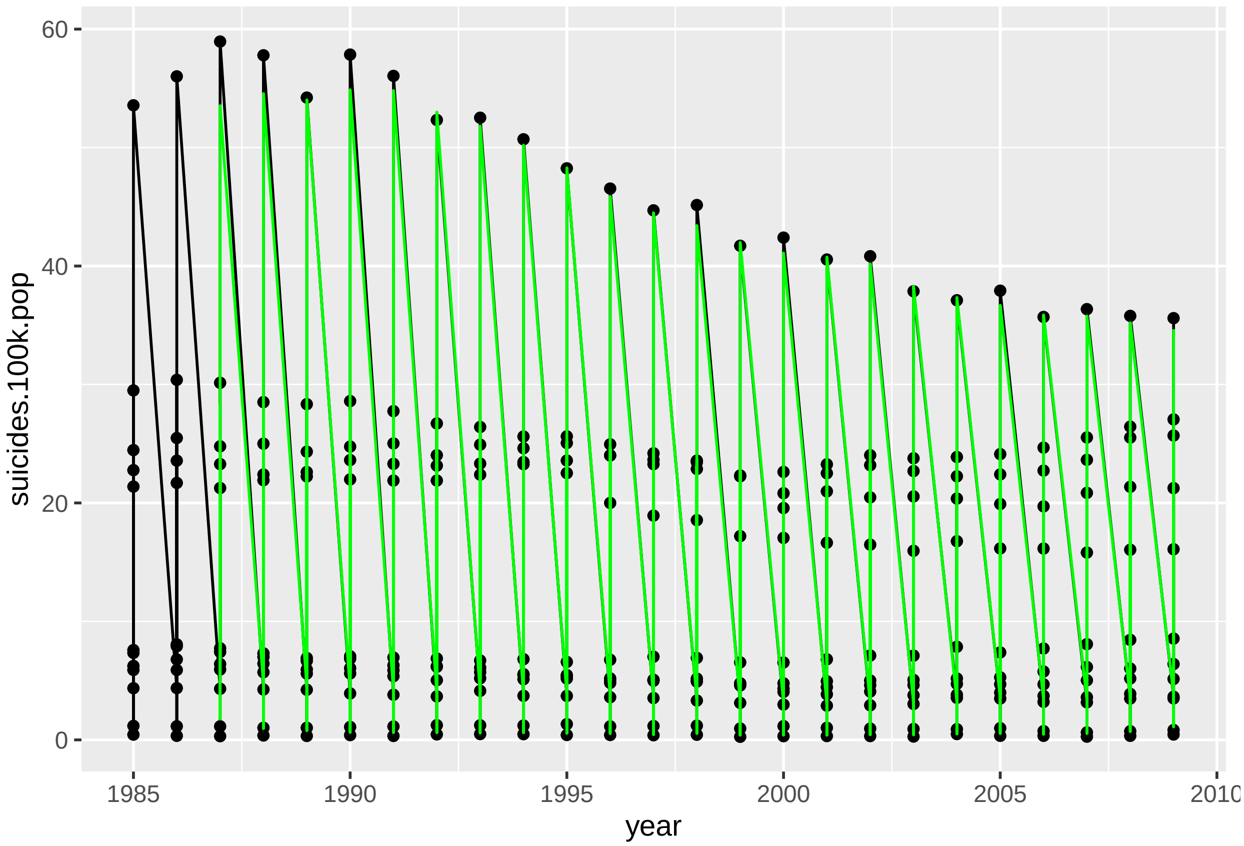

```r
## Trying Random Forest
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
set.seed(349)
```

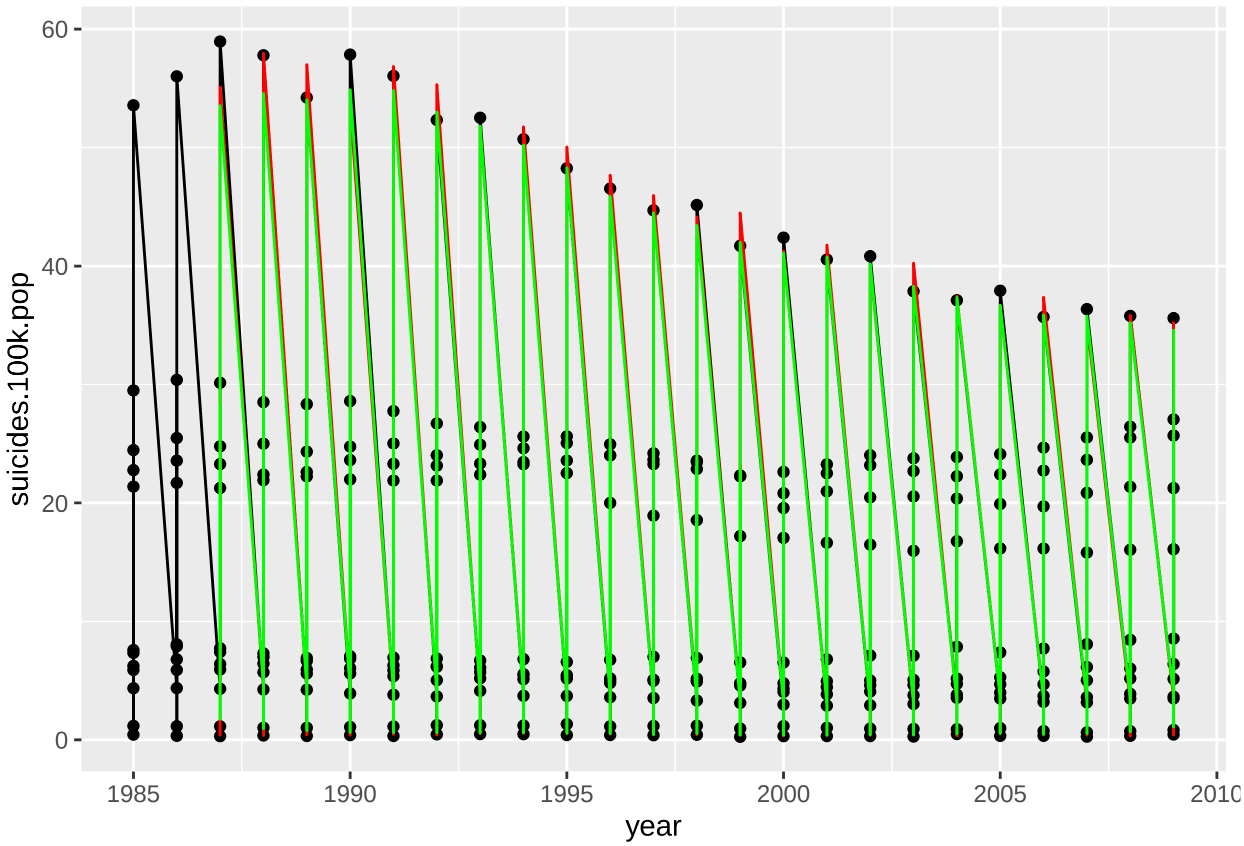
```r
# Plug in all of the variables that we've created
mod.rf <- randomForest(suicides.100k.pop ~ LastYear + TwoYearsAgo + year, data = tail(trainAR_ts, -24))
ggplot(trainAR_ts, aes(x=year, y=suicides.100k.pop)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y=predict(mod.rf, newdata=trainAR_ts)), col="green")
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```




```r
# Both on the same plot:
ggplot(trainAR_ts, aes(x=year, y=suicides.100k.pop)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y=predict(mod2b, newdata=trainAR_ts)), col="red") +
  geom_line(aes(y=predict(mod.rf, newdata=trainAR_ts)), col="green")
```

```
## Warning: Removed 24 rows containing missing values (geom_path).

## Warning: Removed 24 rows containing missing values (geom_path).
```




```r
# Create Test Set
test_ts_final <- test_ts %>%
  mutate(LastYear=c(rep(NA, 12), head(suicides.100k.pop, -12))) %>%
  mutate(TwoYearsAgo = c(rep(NA, 24), head(suicides.100k.pop, -24)))
```

```r
# Test set prediction and OSR^2
pred.test <- predict(mod2b, newdata = test_ts_final)
OSR2(tail(pred.test, -24), trainAR_ts$suicides.100k.pop, tail(test_ts_final$suicides.100k.pop, -24))
```

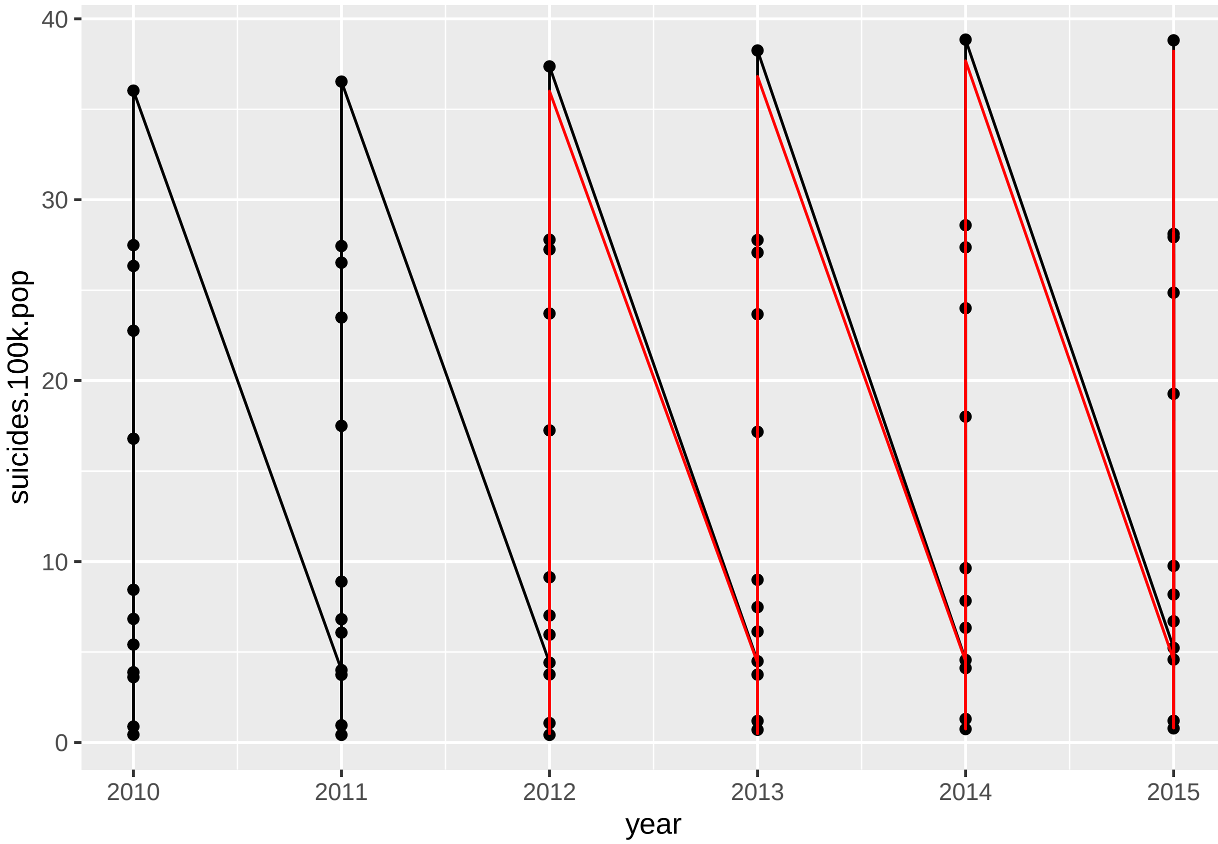
```
## Warning in test - predictions: longer object length is not a multiple of
## shorter object length

## [1] 0.9090414
```

```r
pred.test.rf <- predict(mod.rf, newdata = test_ts_final)
OSR2(tail(pred.test.rf, -24), trainAR_ts$suicides.100k.pop, tail(test_ts_final$suicides.100k.pop, -24))
```

```
## Warning in test - predictions: longer object length is not a multiple of
## shorter object length

## [1] 0.8945664
```

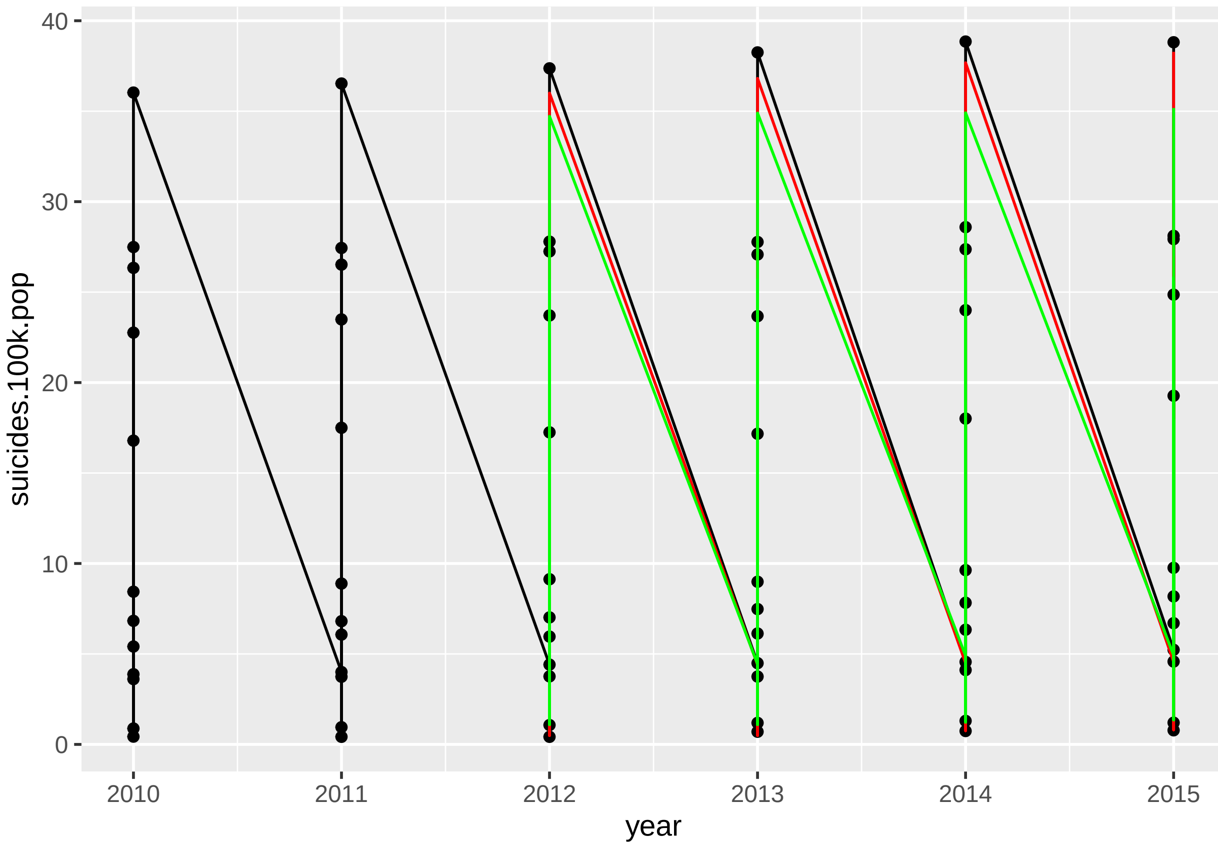
```r
# we should test with a greater fraction in test set or go with random forest maybe?

# Test set plots
ggplot(test_ts_final, aes(x=year, y=suicides.100k.pop)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y=pred.test), col="red")
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```




```r
ggplot(test_ts_final, aes(x=year, y=suicides.100k.pop)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y=pred.test), col="red") +
  geom_line(aes(y=pred.test.rf), col="green")
```

```
## Warning: Removed 24 rows containing missing values (geom_path).

## Warning: Removed 24 rows containing missing values (geom_path).
```


40
30
20
10
0
suicides.100k.pop
2010
2011
2012
2013
2014
2015
year