Regression Analysis of Water Systems

John Meroth

2019-10-23

## DRINKING WATER VIOLATIONS

The following logistic regression analysis in R estimates the probability of a violation occurring based upon the chosen feature vectors. For most choices, it is believed that no correlation will be seen (ROC plot a straight line). In the event a collection of useful features are discovered, the AUC of the ROC plot will increase and the associated model may prove useful in highlighting at-risk water systems.

Notes:  
Sparse data exists prior to 2008, therefore we use data from 2008 and later.  
Only data from currently “Active” water systems are considered.  
Although some systems report violations quarterly, some report only annually.

## ORGANIZING TEST DATA

The EPA’s SDWIS tables WATER\_SYSTEM, VIOLATION, and WATER\_SYSTEM\_FACILITY can be used to generate a set of labelled test data. Each record, identified by “PWSID” represents a water system in the United States.

### Setup the environment

library(dplyr)  
library(dummies)  
library(readr)  
library(tidyverse)  
library(pROC)  
library(lubridate)  
library(effects)

### Import data from pwsid.csv

This file was created by querying the WATER\_SYSTEM table for all active water systems.  
SQL code: SELECT PWSID FROM WATER\_SYSTEM WHERE PWS\_ACTIVITY\_CODE = 'A'

getwd()

## [1] "C:/Users/jmeroth/WATER"

setwd('C:/Users/jmeroth/WATER/derived')  
df <- read\_csv("pwsid.csv")

## Parsed with column specification:  
## cols(  
## PWSID = col\_character()  
## )

summary(df)

## PWSID   
## Length:146055   
## Class :character   
## Mode :character

### Import data from vHealth.csv

This file was created by filtering the VIOLATION table for health-based violations.  
SQL code: SELECT PWSID , cast(VIOLATION\_ID AS TEXT) VIOLATION\_ID , COMPL\_PER\_BEGIN\_DATE FROM VIOLATION WHERE IS\_HEALTH\_BASED\_IND = "Y"

setwd('C:/Users/jmeroth/WATER/derived')  
vH <- read\_csv("vHealth.csv", col\_names = TRUE, cols(VIOLATION\_ID = "c"))  
summary(vH)

## PWSID VIOLATION\_ID COMPL\_PER\_BEGIN\_DATE  
## Length:200718 Length:200718 Length:200718   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character

Convert character field to date:

vH <- vH %>% mutate(COMPL\_PER\_BEGIN\_DATE = as.Date(COMPL\_PER\_BEGIN\_DATE, format = "%d-%B-%y"))  
str(vH)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 200718 obs. of 3 variables:  
## $ PWSID : chr "NJ1413324" "CA5403141" "CA5403141" "CA4300614" ...  
## $ VIOLATION\_ID : chr "13" "1300004" "1200002" "1217002" ...  
## $ COMPL\_PER\_BEGIN\_DATE: Date, format: "2018-09-11" "2013-01-01" ...

Create label field “H\_viol\_2017” from health violation data:

vH <- vH %>%   
 filter(year(vH$COMPL\_PER\_BEGIN\_DATE) == 2017) %>%  
 group\_by(PWSID) %>%  
 summarise(H\_viol\_2017 = 1)  
head(vH)

## # A tibble: 6 x 2  
## PWSID H\_viol\_2017  
## <chr> <dbl>  
## 1 010502003 1  
## 2 055294401 1  
## 3 055294801 1  
## 4 062002156 1  
## 5 063500006 1  
## 6 063500119 1

Merge data into df forming the test data set:

df <- merge(df, vH, by.x="PWSID", by.y="PWSID", all.x=TRUE)  
df[is.na(df)] <- 0  
summary(df)

## PWSID H\_viol\_2017   
## Length:146055 Min. :0.0000   
## Class :character 1st Qu.:0.0000   
## Mode :character Median :0.0000   
## Mean :0.0387   
## 3rd Qu.:0.0000   
## Max. :1.0000

## FEATURE ENGINEERING

“Features” are characteristics or measurments of each sample in our data set. A good feature will be somehow related to the chances of a system registering a violation. The exact relationship to a violation need not be know for the feature to be useful.

### TREATED\_SOURCE

Add a binary predictor variable, “TREATED\_SOURCE” (1 if treatment facility exists). The file treated\_source.csv was created using this SQL code:  
with active\_facilities AS ( select ws.PWSID from WATER\_SYSTEM2 ws left join WATER\_SYSTEM\_FACILITY\_3 fa on ws.PWSID = fa.PWSID where ws.PWS\_ACTIVITY\_CODE = ‘A’ and fa.IS\_SOURCE\_TREATED\_IND = ‘Y’ and fa.FACILITY\_ACTIVITY\_CODE = ‘A’) Select DISTINCT PWSID from active\_facilities

setwd("C:/Users/jmeroth/water/derived")  
treated\_source <- read\_csv("treated\_source.csv")

## Parsed with column specification:  
## cols(  
## PWSID = col\_character()  
## )

treated\_source$TREATED\_SOURCE = 1  
df <- merge(df, treated\_source, by.x="PWSID", by.y="PWSID"  
 , all.x=TRUE)  
df[is.na(df)] <- 0

### VIOLATION\_CATEGORY\_CODE

Add a group of variables indicating the occurrance of a non-health violation in the prior year (2016). The file violation\_category\_code.csv was created using this SQL code:  
SELECT PWSID, VIOLATION\_CATEGORY\_CODE, substr(COMPL\_PER\_BEGIN\_DATE, -2) AS vYear FROM VIOLATION2 WHERE IS\_HEALTH\_BASED\_IND = “N” and PWS\_ACTIVITY\_CODE = ‘A’ and vYear = ‘16’ GROUP by PWSID, VIOLATION\_CATEGORY\_CODE, vYear  
The resulting records were pivoted by PWSID on VIOLATION\_CATEGORY\_CODE.

setwd("C:/Users/jmeroth/water/derived")  
violation\_category\_code <- read\_csv("violation\_category\_code.csv")

## Parsed with column specification:  
## cols(  
## PWSID = col\_character(),  
## MON = col\_double(),  
## MR = col\_double(),  
## Other = col\_double(),  
## RPT = col\_double()  
## )

df <- merge(df, violation\_category\_code, by.x="PWSID", by.y="PWSID"  
 , all.x=TRUE)  
df[is.na(df)] <- 0

### PRIMARY\_SOURCE\_CODE

The file primary\_source\_code.csv was created using this SQL code:  
SELECT PWSID, PRIMARY\_SOURCE\_CODE FROM WATER\_SYSTEM WHERE PWS\_ACTIVITY\_CODE = ‘A’

Add a categorical variable “PRIMARY\_SOURCE\_CODE”:

setwd("C:/Users/jmeroth/water/derived")  
psc <- read\_csv("primary\_source\_code.csv")

## Parsed with column specification:  
## cols(  
## PWSID = col\_character(),  
## PRIMARY\_SOURCE\_CODE = col\_character()  
## )

# Merge the new feature vector:  
df <- merge(df, psc, by.x="PWSID", by.y="PWSID", all.x=TRUE)  
# Convert empty cells to zero:  
df[is.na(df)] <- 0

### PWS\_TYPE\_CODE

The file pws\_type\_code.csv was created using this SQL code:  
*SELECT PWSID, PWS\_TYPE\_CODE FROM WATER\_SYSTEM WHERE PWS\_ACTIVITY\_CODE = ‘A’*

Add a categorical variable “PWS\_TYPE\_CODE”:

setwd("C:/Users/jmeroth/water/derived")  
ptc <- read\_csv("pws\_type\_code.csv")

## Parsed with column specification:  
## cols(  
## PWSID = col\_character(),  
## PWS\_TYPE\_CODE = col\_character()  
## )

# Merge the new feature vector:  
df <- merge(df, ptc, by.x="PWSID", by.y="PWSID", all.x=TRUE)  
# Convert empty cells to zero:  
df[is.na(df)] <- 0

### PRIOR YEAR TOTAL RAINFALL

The file wsAnnualRain.csv was created using the Jupyter Notebook: match\_water\_weather.ipynb (from Mia Polansky, June 20, 2019) This file contains the geolocated watersystems along with the annual rainfall at the nearest weather station.

Add a normalized, continuous variable “r2016n”:

# Read in the rainfall data:  
setwd("C:/Users/jmeroth/WATER/Weather")  
rain <- read\_csv("wsAnnualRain.csv")

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## X1 = col\_double(),  
## PWSID = col\_character(),  
## `0` = col\_double()  
## )

# Normalize the rainfall data:  
rain$X1 <- NULL  
colnames(rain)[colnames(rain)=="0"] <- "r2016"  
rmin <- min(rain$r2016)  
rmax <- max(rain$r2016)  
rscale = rmax-rmin  
rain["r2016n"] <- (rain$r2016-rmin)/rscale - 1.0  
rain$r2016 <- NULL

# Merge the new feature vector:  
df <- merge(df, rain, by.x="PWSID", by.y="PWSID", all.x=TRUE)  
# Convert empty cells to zero:  
df[is.na(df)] <- 0

## MODELLING

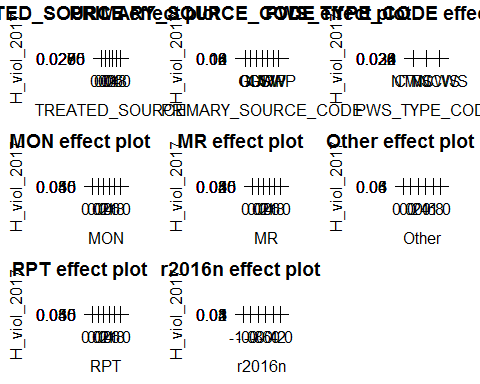
### Run the Logistic Regression Model:

logit <- glm(H\_viol\_2017 ~ TREATED\_SOURCE + PRIMARY\_SOURCE\_CODE + PWS\_TYPE\_CODE + MON + MR + Other + RPT + r2016n  
 , data = df, family = "binomial")  
summary(logit)

##   
## Call:  
## glm(formula = H\_viol\_2017 ~ TREATED\_SOURCE + PRIMARY\_SOURCE\_CODE +   
## PWS\_TYPE\_CODE + MON + MR + Other + RPT + r2016n, family = "binomial",   
## data = df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4558 -0.2811 -0.2218 -0.2010 3.4678   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -6.10756 1.01779 -6.001 1.96e-09 \*\*\*  
## TREATED\_SOURCE -0.00962 0.02943 -0.327 0.744   
## PRIMARY\_SOURCE\_CODEGU 1.34257 1.01689 1.320 0.187   
## PRIMARY\_SOURCE\_CODEGUP 0.28479 1.13184 0.252 0.801   
## PRIMARY\_SOURCE\_CODEGW 0.50807 1.01055 0.503 0.615   
## PRIMARY\_SOURCE\_CODEGWP 0.31298 1.01509 0.308 0.758   
## PRIMARY\_SOURCE\_CODESW 1.80684 1.01137 1.787 0.074 .   
## PRIMARY\_SOURCE\_CODESWP 1.04631 1.01136 1.035 0.301   
## PWS\_TYPE\_CODENTNCWS -0.22593 0.04802 -4.704 2.55e-06 \*\*\*  
## PWS\_TYPE\_CODETNCWS -0.40118 0.03369 -11.908 < 2e-16 \*\*\*  
## MON 0.68047 0.04173 16.305 < 2e-16 \*\*\*  
## MR 0.79812 0.03390 23.546 < 2e-16 \*\*\*  
## Other 0.84495 0.03693 22.880 < 2e-16 \*\*\*  
## RPT 0.60353 0.08020 7.525 5.27e-14 \*\*\*  
## r2016n -2.63473 0.14045 -18.759 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 47843 on 146054 degrees of freedom  
## Residual deviance: 43571 on 146040 degrees of freedom  
## AIC: 43601  
##   
## Number of Fisher Scoring iterations: 7

Result of regression using the “effects” library:

plot(effects::allEffects(logit), axes=list(grid=FALSE))



### Create Test Data

test\_data <- df  
test\_data$PWSID <- NULL  
test\_data$H\_viol\_2017 <- NULL

Calculate probabilities P, add to data frame.

df["P"] <- predict(logit, test\_data, type="response")

## ANALYSIS

### Analysis of Means

df %>% group\_by(H\_viol\_2017) %>% summarise(mean = mean(P))

## # A tibble: 2 x 2  
## H\_viol\_2017 mean  
## <dbl> <dbl>  
## 1 0 0.0370  
## 2 1 0.0797

### Pivot tables

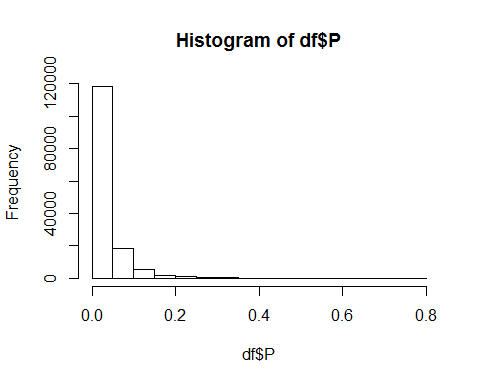
xtabs(~H\_viol\_2017 + PRIMARY\_SOURCE\_CODE, data = df)

## PRIMARY\_SOURCE\_CODE  
## H\_viol\_2017 0 GU GUP GW GWP SW SWP  
## 0 60 882 131 123724 3441 4382 7783  
## 1 1 92 4 4165 110 712 568

xtabs(~H\_viol\_2017 + PWS\_TYPE\_CODE, data = df)

## PWS\_TYPE\_CODE  
## H\_viol\_2017 CWS NTNCWS TNCWS  
## 0 46763 16959 76681  
## 1 2915 599 2138

### Probability Distribution



### Confusion Marix with threshold set as P = 0.3

df %>% group\_by(P > 0.3) %>% filter(H\_viol\_2017 == 1) %>% summarise(n())

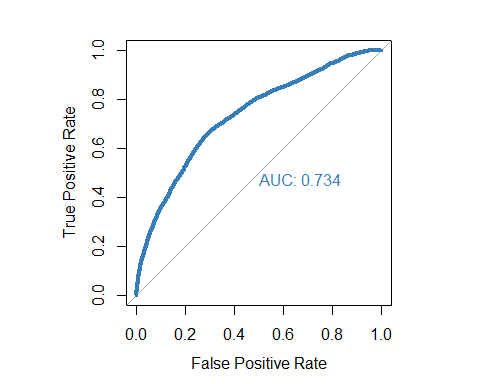
## # A tibble: 2 x 2  
## `P > 0.3` `n()`  
## <lgl> <int>  
## 1 FALSE 5499  
## 2 TRUE 153

df %>% group\_by(P > 0.3) %>% filter(H\_viol\_2017 == 0) %>% summarise(n())

## # A tibble: 2 x 2  
## `P > 0.3` `n()`  
## <lgl> <int>  
## 1 FALSE 140028  
## 2 TRUE 375

### ROC

#Create a square plot  
par(pty = "s")  
roc(df$H\_viol\_2017, df$P, plot=TRUE, legacy.axes=TRUE, percent=FALSE, xlab="False Positive Rate", ylab="True Positive Rate", col="#377eb8", lwd=4, print.auc=TRUE)



##   
## Call:  
## roc.default(response = df$H\_viol\_2017, predictor = df$P, percent = FALSE, plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Rate", ylab = "True Positive Rate", col = "#377eb8", lwd = 4, print.auc = TRUE)  
##   
## Data: df$P in 140403 controls (df$H\_viol\_2017 0) < 5652 cases (df$H\_viol\_2017 1).  
## Area under the curve: 0.7341

## FUTURE WORK

Two years of rainfall data should be included as well as demographic data.