DETECTING FLU EPIDEMICS VIA SEARCH ENGINE QUERY DATA

Anuranjan

25 September 2016

Load data sets

```
FluTrain=read.csv("FluTrain.csv")
FluTest=read.csv("FluTest.csv")
summary(FluTrain)
```

```
##                       Week          ILI            Queries
##  2004-01-04 - 2004-01-10:  1   Min.   :0.5341   Min.   :0.04117
##  2004-01-11 - 2004-01-17:  1   1st Qu.:0.9025   1st Qu.:0.15671
##  2004-01-18 - 2004-01-24:  1   Median :1.2526   Median :0.28154
##  2004-01-25 - 2004-01-31:  1   Mean   :1.6769   Mean   :0.28603
##  2004-02-01 - 2004-02-07:  1   3rd Qu.:2.0587   3rd Qu.:0.37849
##  2004-02-08 - 2004-02-14:  1   Max.   :7.6189   Max.   :1.00000
##  (Other)                :411
```

```
str(FluTrain)
```

```
## 'data.frame':    417 obs. of  3 variables:
##  $ Week   : Factor w/ 417 levels "2004-01-04 - 2004-01-10",..: 1 2 3 4 5 6
7 8 9 10 ...
##  $ ILI    : num  2.42 1.81 1.71 1.54 1.44 ...
##  $ Queries: num  0.238 0.22 0.226 0.238 0.224 ...
```

Which week corresponds to the highest percentage of ILI-related physician visits?

```
FluTrain$Week[which.max(FluTrain$ILI)]
```

```
## [1] 2009-10-18 - 2009-10-24
## 417 Levels: 2004-01-04 - 2004-01-10 ... 2011-12-25 - 2011-12-31
```

Which week corresponds to the highest percentage of ILI-related query fraction?

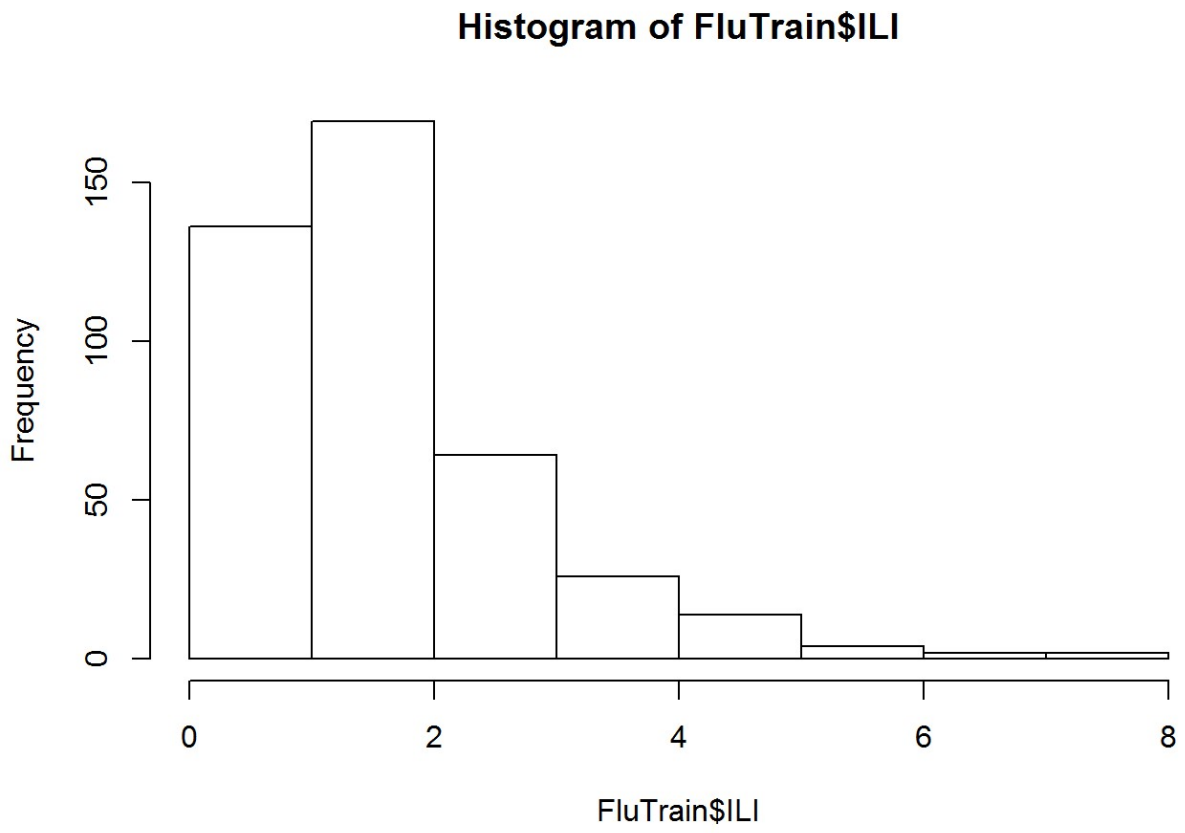
```
FluTrain$Week[which.max(FluTrain$Queries)]
```

```
## [1] 2009-10-18 - 2009-10-24
## 417 Levels: 2004-01-04 - 2004-01-10 ... 2011-12-25 - 2011-12-31
```

We see that both are coming in the same week. It may mean they are related.

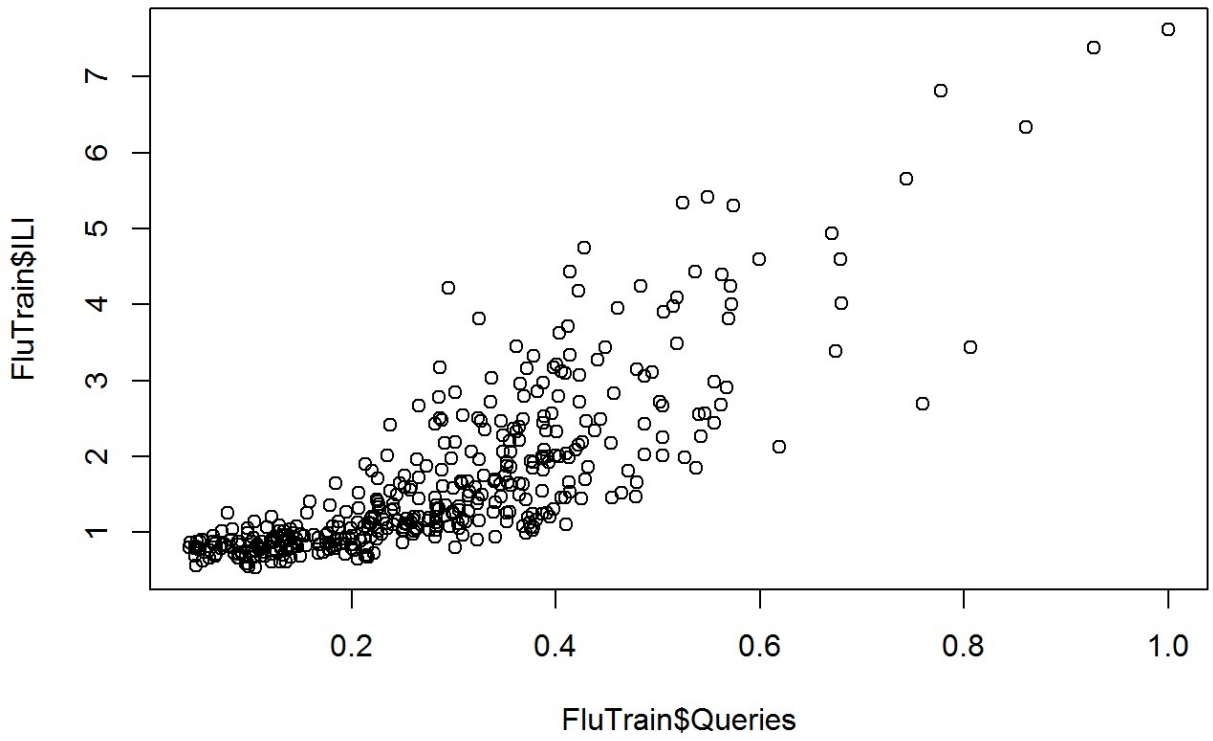
Histogram for ILI

```
hist(FluTrain$ILI)
```

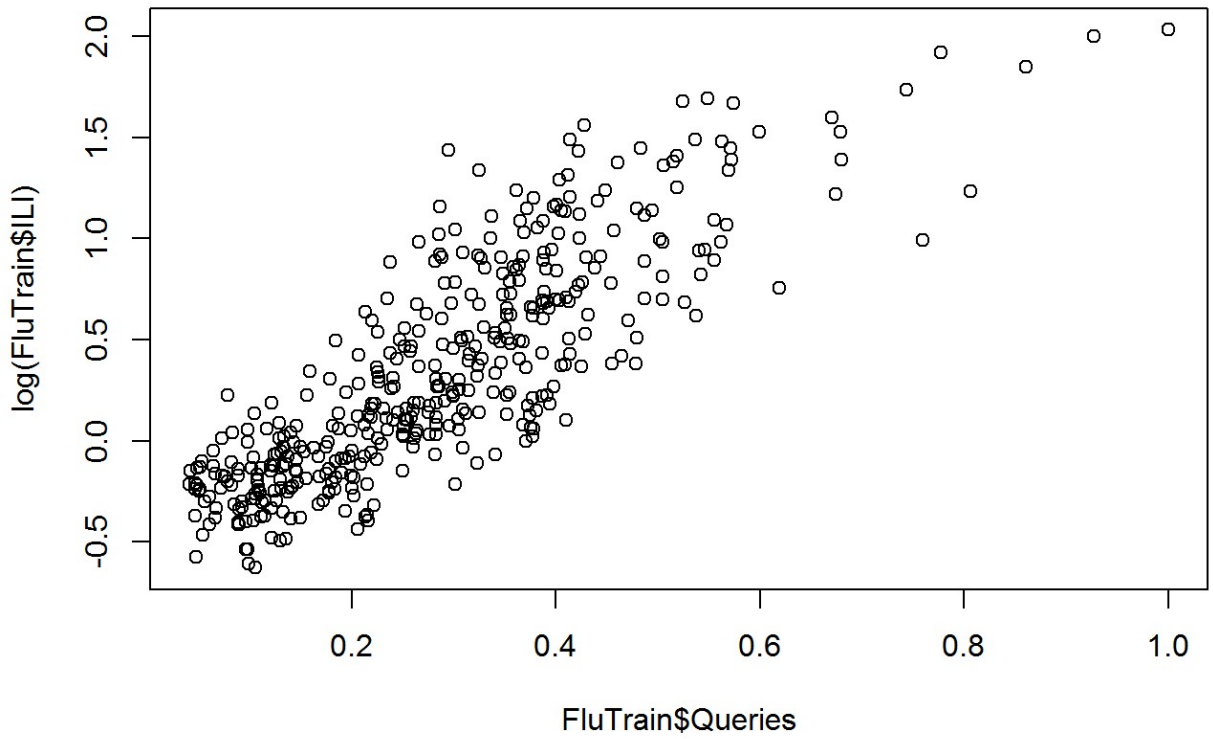


As the graph shows, data is right skewed. **When handling a skewed dependent variable, it is often useful to predict the logarithm of the dependent variable instead of the dependent variable itself – this prevents the small number of unusually large or small observations from having an undue influence on the sum of squared errors of predictive models. In this problem, we will predict the natural log of the ILI variable, which can be computed in R using the log() function.**

```
plot(FluTrain$Queries,FluTrain$ILI)
```




```
plot(FluTrain$Queries,log(FluTrain$ILI))
```



Visually, there is more positive, linear relationship between log(ILI) and Queries.

Building Linear Model.

We are using log of dependent variable in the model.

```
FluTrend1 = lm(log(ILI)~Queries, data=FluTrain)
summary(FluTrend1)
```

```
##
## Call:
## lm(formula = log(ILI) ~ Queries, data = FluTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76003 -0.19696 -0.01657  0.18685  1.06450
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.49934    0.03041  -16.42   <2e-16 ***
## Queries      2.96129    0.09312   31.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2995 on 415 degrees of freedom
## Multiple R-squared:  0.709,  Adjusted R-squared:  0.7083
## F-statistic:  1011 on 1 and 415 DF,  p-value: < 2.2e-16
```

Corelation

```
cor(FluTrain$Queries,log(FluTrain$ILI))
```

```
## [1] 0.8420333
```

```
summary(FluTrend1)$r.squared/cor(FluTrain$Queries,log(FluTrain$ILI))
```

```
## [1] 0.8420333
```

R square value is not stored in Model variable like residuals and coefficients. To access it, use summary(model)$r.squared

For a single variable linear regression model, there is a direct relationship between the R-squared and the correlation between the independent and the dependent variables: Correlation^2 is equal to the R-squared value. It can be proved that this is always the case.

Prediction

```
PredTest1=exp(predict(FluTrend1,newdata = FluTest))
```

The dependent variable in our model is log(ILI). Converting from predictions of log(ILI) to predictions of ILI via exponentiation, or the exp() function.

Estimate for the percentage of ILI-related physician visits for the week of March 11, 2012?

```
temp_pred = PredTest1[which(FluTest$Week == "2012-03-11 - 2012-03-17")]
temp_pred
```

```
##       11
## 2.187378
```

We could have just output FluTest$Week to find which element corresponds to March 11, 2012 , however, **which** function gives answer in just one line.

What is the relative error betweeen the estimate (our prediction) and the observed value for the week of March 11, 2012?

```
temp_test = FluTest$ILI[which(FluTest$Week == "2012-03-11 - 2012-03-17")]
(temp_test - temp_pred)/temp_test
```

```
##         11
## 0.04623827
```

Root Mean Square Error (RMSE)

```
SSE = (PredTest1 - FluTest$ILI) ^ 2
# RMSE = sqrt(SSE/nrow(FluTest))
RMSE1 = sqrt(mean((PredTest1 - FluTest$ILI) ^ 2))
RMSE1
```

```
## [1] 0.7490645
```

The observations in this dataset are consecutive weekly measurements of the dependent and independent variables. This sort of dataset is called a “time series.” Often, statistical models can be improved by predicting the current value of the dependent variable using the value of the dependent variable from earlier weeks. In our models, this means we will predict the ILI variable in the current week using values of the ILI variable from previous weeks.

First, we need to decide the amount of time to lag the observations. Because the ILI variable is reported with a 1- or 2-week lag, a decision maker cannot rely on the previous week’s ILI value to predict the current week’s value. Instead, the decision maker will only have data available from 2 or more weeks ago. We will build a variable called ILILag2 that contains the ILI value from 2 weeks before the current observation.

To do so, we will use the “zoo” package, which provides a number of helpful methods for time series models.

```
# install.packages("zoo")
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
ILILag2 = lag(zoo(FluTrain$ILI), -2, na.pad=TRUE)
FluTrain$ILILag2 = coredata(ILILag2)
ILILag2 = lag(zoo(FluTest$ILI), -2, na.pad=TRUE)
FluTest$ILILag2 = coredata(ILILag2)
```

In these commands, the value of -2 passed to lag means to return 2 observations before the current one; a positive value would have returned future observations. The parameter na.pad=TRUE means to add missing values for the first two weeks of our dataset, where we can’t compute the data from 2 weeks earlier.

** values missing in the new ILILag2 variable? **

```
table(is.na(FluTrain$ILILag2))
```

```
##
## FALSE  TRUE
##   415     2
```

```
table(is.na(FluTest$ILILag2))
```

```
##
## FALSE  TRUE
##    50     2
```

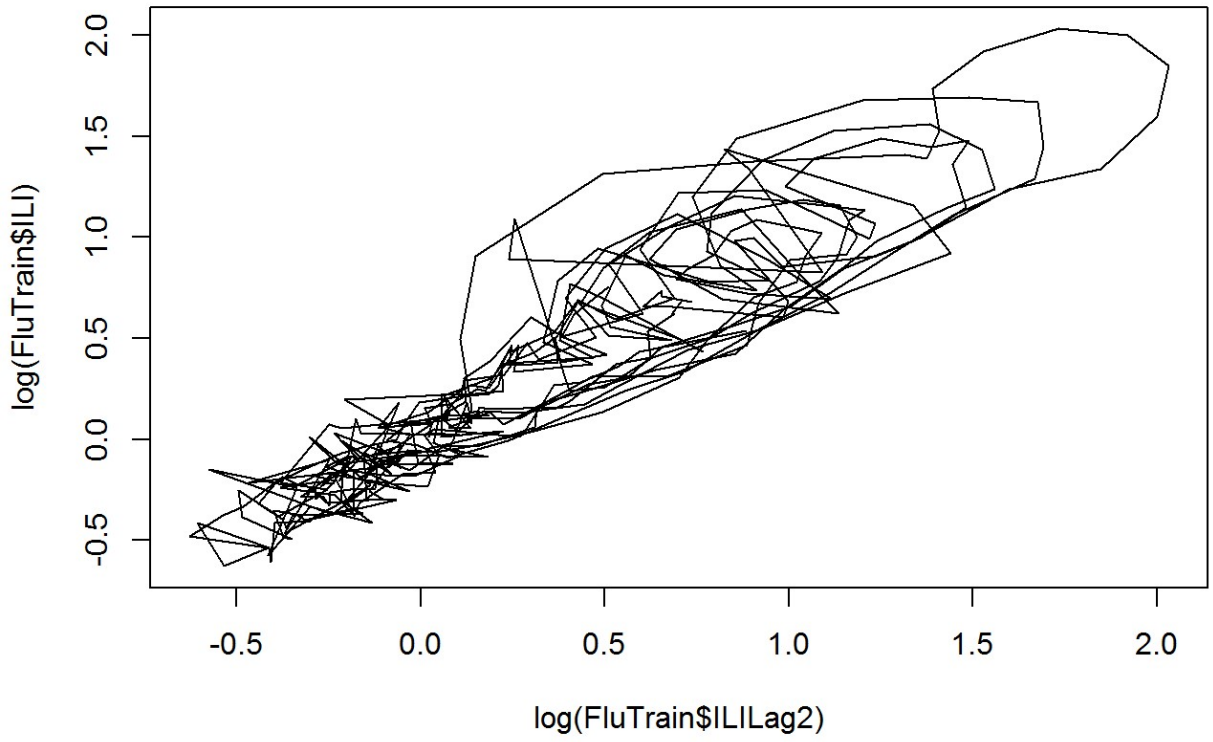
In this problem, the training and testing sets are split sequentially – the training set contains all observations from 2004-2011 and the testing set contains all observations from 2012. There is no time gap between the two datasets, meaning the first observation in FluTest was recorded one week after the last observation in FluTrain. The ILI value of the second-to-last and last observation in the FluTrain data frame can be used to fill first two ILILag2 values in FluTest dataset.

```
FluTest$ILILag2[1] = FluTrain$ILI[nrow(FluTrain)-1]
FluTest$ILILag2[2] = FluTrain$ILI[nrow(FluTrain)]
head(FluTest)
```

```
##                      Week      ILI   Queries  ILILag2
## 1 2012-01-01 - 2012-01-07 1.766707 0.5936255 1.852736
## 2 2012-01-08 - 2012-01-14 1.543401 0.4993360 2.124130
## 3 2012-01-15 - 2012-01-21 1.647615 0.5006640 1.766707
## 4 2012-01-22 - 2012-01-28 1.684297 0.4794157 1.543401
## 5 2012-01-29 - 2012-02-04 1.863542 0.4714475 1.647615
## 6 2012-02-05 - 2012-02-11 1.864079 0.5033201 1.684297
```

** log of ILILag2 vs log of ILI**

```
plot(log(FluTrain$ILILag2),log(FluTrain$ILI),type = "l")
```



From plot(log(FluTrain$ILILag2), $log(FluTrain$ ILI)), we observe a strong positive relationship.

Training a Time Series Model

```
FluTrend2 = lm(log(ILI)~Queries+log(ILILag2), data=FluTrain)
```

Here ILILag2 is also being used in the model.

```
summary(FluTrend2)
```

```
##
## Call:
## lm(formula = log(ILI) ~ Queries + log(ILILag2), data = FluTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52209 -0.11082 -0.01819  0.08143  0.76785
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.24064    0.01953  -12.32   <2e-16 ***
## Queries       1.25578    0.07910   15.88   <2e-16 ***
## log(ILILag2)  0.65569    0.02251   29.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1703 on 412 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.9063, Adjusted R-squared:  0.9059
## F-statistic:  1993 on 2 and 412 DF,  p-value: < 2.2e-16
```

Moving from FluTrend1 to FluTrend2, in-sample R^2 improved from 0.709 to 0.9063, and the new variable is highly significant. As a result, there is no sign of overfitting, and FluTrend2 is superior to FluTrend1 on the training set.

Prediction and RMSE

```
PredTest2=exp(predict(FluTrend2,newdata = FluTest))
RMSE2 = sqrt(mean((PredTest2 - FluTest$ILI)^2))
RMSE2
```

```
## [1] 0.2942029
```

We can see that RMSE2 (0.2942029) has much lower value than RMSE1(0.7490645)

In this problem, we used a simple time series model with a single lag term. **ARIMA** models are a more general form of the model we built, which can include multiple lag terms as well as more complicated combinations of previous values of the dependent variable.

****