# R Notebook

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.2
```

```
## -- Attaching packages ------------------------------------------------------------------------------
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.0     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
## Warning: package 'tibble' was built under R version 3.4.4
```

```
## Warning: package 'tidyr' was built under R version 3.4.3
```

```
## Warning: package 'purrr' was built under R version 3.4.4
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
## Warning: package 'forcats' was built under R version 3.4.3
```

```
## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 3.4.4
```

Flu epidemics constitute a major public health concern causing respiratory illnesses, hospitalizations, and deaths. According to the National Vital Statistics Reports published in October 2012, influenza ranked as the eighth leading cause of death in 2011 in the United States. Each year, 250,000 to 500,000 deaths are attributed to influenza related diseases throughout the world.

The U.S. Centers for Disease Control and Prevention (CDC) and the European Influenza Surveillance Scheme (EISS) detect influenza activity through virologic and clinical data, including Influenza-like Illness (ILI) physician visits. Reporting national and regional data, however, are published with a 1-2 week lag.

The Google Flu Trends project was initiated to see if faster reporting can be made possible by considering flu-related online search queries – data that is available almost immediately.

### 1.1) Understanding the Data

**We would like to estimate influenza-like illness (ILI) activity using Google web search logs**. Fortunately, one can easily access this data online:

ILI Data - The CDC publishes on its website the official regional and state-level percentage of patient visits to healthcare providers for ILI purposes on a weekly basis.

Google Search Queries - Google Trends allows public retrieval of weekly counts for every query searched by users around the world. For each location, the counts are normalized by dividing the count for each query in a particular week by the total number of online search queries submitted in that location during the week. Then, the values are adjusted to be between 0 and 1.

The csv file FluTrain.csv aggregates this data from January 1, 2004 until December 31, 2011 as follows:

**Week -** The range of dates represented by this observation, in year/month/day format.

**ILI** - This column lists the percentage of ILI-related physician visits for the corresponding week.

**Queries** - This column lists the fraction of queries that are ILI-related for the corresponding week, adjusted to be between 0 and 1 (higher values correspond to more ILI-related search queries).

Before applying analytics tools on the training set, we first need to understand the data at hand. Load “FluTrain.csv” into a data frame called FluTrain. Looking at the time period 2004-2011, which week corresponds to the highest percentage of ILI-related physician visits? Select the day of the month corresponding to the start of this week.

```
flu_train = read.csv('FluTrain.csv')

flu_train %>% glimpse()
```

```
## Observations: 417
## Variables: 3
## $ Week    <fct> 2004-01-04 - 2004-01-10, 2004-01-11 - 2004-01-17, 2004...
## $ ILI     <dbl> 2.4183312, 1.8090560, 1.7120239, 1.5424951, 1.4378683,...
## $ Queries <dbl> 0.23771580, 0.22045153, 0.22576361, 0.23771580, 0.2244...
```

```
flu_train%>%
  count(ILI, Week) %>%
  arrange(desc(ILI)) # Oct 18, 2009
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
## # A tibble: 417 x 3
##      ILI Week                        n
##    <dbl> <fct>                   <int>
##  1  7.62 2009-10-18 - 2009-10-24     1
##  2  7.39 2009-10-25 - 2009-10-31     1
##  3  6.82 2009-10-11 - 2009-10-17     1
##  4  6.34 2009-11-01 - 2009-11-07     1
##  5  5.66 2009-10-04 - 2009-10-10     1
##  6  5.42 2008-02-10 - 2008-02-16     1
##  7  5.35 2008-02-03 - 2008-02-09     1
##  8  5.30 2008-02-17 - 2008-02-23     1
##  9  4.94 2009-11-08 - 2009-11-14     1
## 10  4.75 2005-02-13 - 2005-02-19     1
## # ... with 407 more rows
```

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

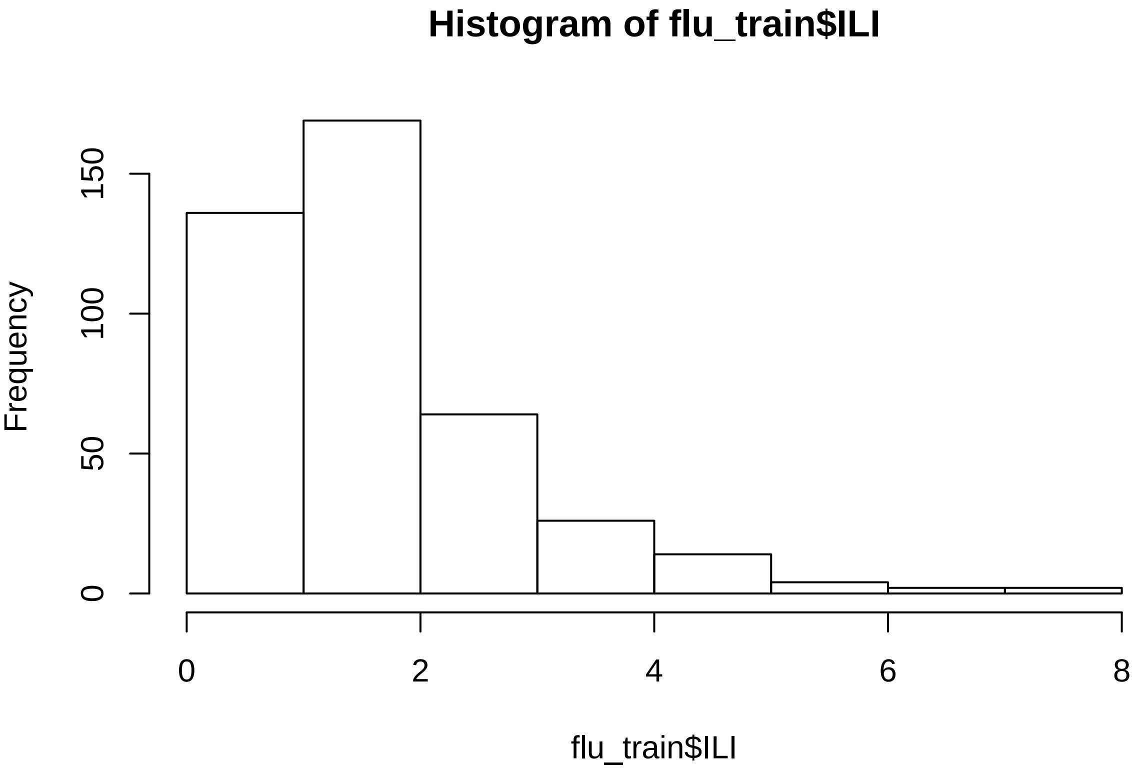
```
flu_train%>%
  count(Queries, Week) %>%
  arrange(desc(Queries)) # Oct 18, 2009
```

```
## # A tibble: 417 x 3
##    Queries Week                        n
##      <dbl> <fct>                   <int>
##  1   1     2009-10-18 - 2009-10-24     1
##  2   0.927 2009-10-25 - 2009-10-31     1
##  3   0.861 2009-11-01 - 2009-11-07     1
##  4   0.806 2010-12-26 - 2011-01-01     1
##  5   0.777 2009-10-11 - 2009-10-17     1
##  6   0.760 2011-01-02 - 2011-01-08     1
##  7   0.744 2009-10-04 - 2009-10-10     1
##  8   0.680 2009-09-20 - 2009-09-26     1
## 9    0.679 2009-09-27 - 2009-10-03     1
## 10   0.675 2010-12-19 - 2010-12-25     1
## # ... with 407 more rows
```

### 1.2) Understanding the Data

Let us now understand the data at an aggregate level. Plot the histogram of the dependent variable, ILI. What best describes the distribution of values of ILI?

```
hist(flu_train$ILI)
```

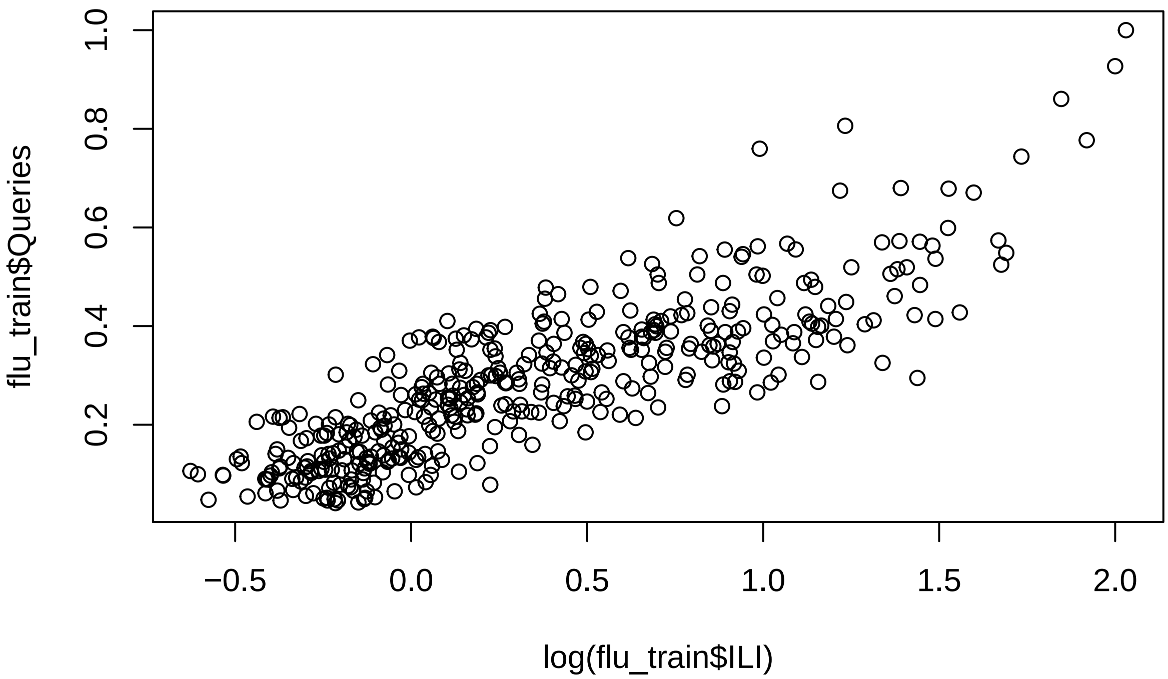


### 1.3) Understanding the Data

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 the natural logarithm of ILI versus Queries. What does the plot suggest?

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



### 2.1) Linear Regression Model

Based on the plot we just made, it seems that a linear regression model could be a good modeling choice. Based on our understanding of the data from the previous subproblem, which model best describes our estimation problem?

```
test1 = lm(ILI~ Queries, data = flu_train)
summary(test1)
```

```
##
## Call:
## lm(formula = ILI ~ Queries, data = flu_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73911 -0.38816 -0.04161  0.31012  2.48517
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01374    0.06646   0.207    0.836
## Queries      5.81454    0.20352  28.570   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6546 on 415 degrees of freedom
## Multiple R-squared:  0.6629, Adjusted R-squared:  0.6621
## F-statistic: 816.2 on 1 and 415 DF,  p-value: < 2.2e-16
```

```
test2 = lm(Queries ~ ILI, data = flu_train)
summary(test2)
```

```
##
## Call:
```

```
## lm(formula = Queries ~ ILI, data = flu_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28038 -0.06798 -0.00700  0.05673  0.35792
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.094842   0.008058   11.77   <2e-16 ***
## ILI         0.114014   0.003991   28.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09166 on 415 degrees of freedom
## Multiple R-squared:  0.6629, Adjusted R-squared:  0.6621
## F-statistic: 816.2 on 1 and 415 DF,  p-value: < 2.2e-16
```

**This is the one**

```
test3 = lm(log(ILI) ~ Queries, data = flu_train)
summary(test3)
```

```
##
## Call:
## lm(formula = log(ILI) ~ Queries, data = flu_train)
##
## 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
```

```
test4 = lm(Queries ~ log(ILI), data = flu_train)
summary(test4)
```

```
##
## Call:
## lm(formula = Queries ~ log(ILI), data = flu_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25232 -0.05899 -0.00435  0.04232  0.31978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.202786   0.004924   41.18   <2e-16 ***
## log(ILI)    0.239429   0.007529   31.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08517 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
```

### 2.2) Linear Regression Model

```r
flu_trend_1 = lm(log(ILI) ~ Queries, data = flu_train)
summary(flu_trend_1)
```

```
##
## Call:
## lm(formula = log(ILI) ~ Queries, data = flu_train)
##
## 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
```

### 2.3) Linear Regression Model

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. What is the relationship we infer from our problem? (Don't forget that you can use the cor function to compute the correlation between two variables.)

```r
val = log(flu_train$ILI)

cor(val, flu_train$Queries)
```

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

```r
0.8420333 ^ 2 #R-squared = Correlation ^ 2
```

```
## [1] 0.7090201
```

### 3.1) Performance on the Test Set

The csv file FluTest.csv provides the 2012 weekly data of the ILI-related search queries and the observed weekly percentage of ILI-related physician visits. Load this data into a data frame called FluTest.

```r
flu_test = read.csv('FluTest.csv')
```

Normally, we would obtain test-set predictions from the model FluTrend1 using the code

PredTest1 = predict(FluTrend1, newdata=FluTest)

However, the dependent variable in our model is log(ILI), so PredTest1 would contain predictions of the log(ILI) value. We are instead interested in obtaining predictions of the ILI value. We can convert from predictions of log(ILI) to predictions of ILI via exponentiation, or the exp() function. The new code, which predicts the ILI value, is

```
pred_test = exp(predict(flu_trend_1, newdata = flu_test))
```

What is our estimate for the percentage of ILI-related physician visits for the week of March 11, 2012? (HINT: You can either just output FluTest$Week to find which element corresponds to March 11, 2012, or you can use the "which" function in R. To learn more about the which function, type ?which in your R console.)

```
which(flu_test$Week == "2012-03-11 - 2012-03-17")
```

```
## [1] 11
```

```
head(pred_test,11)
```

```
##        1        2        3        4        5        6        7        8 
## 3.520332 2.662689 2.673181 2.510160 2.451624 2.694289 2.780402 2.673181 
##        9       10       11 
## 2.375693 2.357081 2.187378
```

### 3.2) Performance on the Test Set

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

(Observed ILI - Estimated ILI)/Observed ILI

```
obs = flu_test[flu_test$Week == "2012-03-11 - 2012-03-17", 'ILI']

est = pred_test[11]

rel_error = (obs - est) / obs
```

### 3.3) Performance on the Test Set ???

What is the Root Mean Square Error (RMSE) between our estimates and the actual observations for the percentage of ILI-related physician visits, on the test set?

```
pred_test = exp(predict(flu_trend_1, newdata = flu_test))

SSE_test_1 = sum((pred_test - flu_test$ILI)^2)

RMSE_1 = sqrt(SSE_test_1 / nrow(flu_test))

RMSE_1
```

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

### 4.1) Training a Time Series Model

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. While many functions are built into R, you need to add new packages to use some functions. New packages can be installed and loaded easily in R, and we will do this many times in this class. Run the following two commands to install and load the zoo package. In the first command, you will be prompted to select a CRAN mirror to use for your download. Select a mirror near you geographically.

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 3.4.3

##
## Attaching package: 'zoo'

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

After installing and loading the zoo package, run the following commands to create the ILILag2 variable in the training set:

```
ILI_lag2 = stats::lag(zoo(flu_train$ILI), -2, na.pad = TRUE)

flu_train$ILI_lag2 = coredata(ILI_lag2)
```

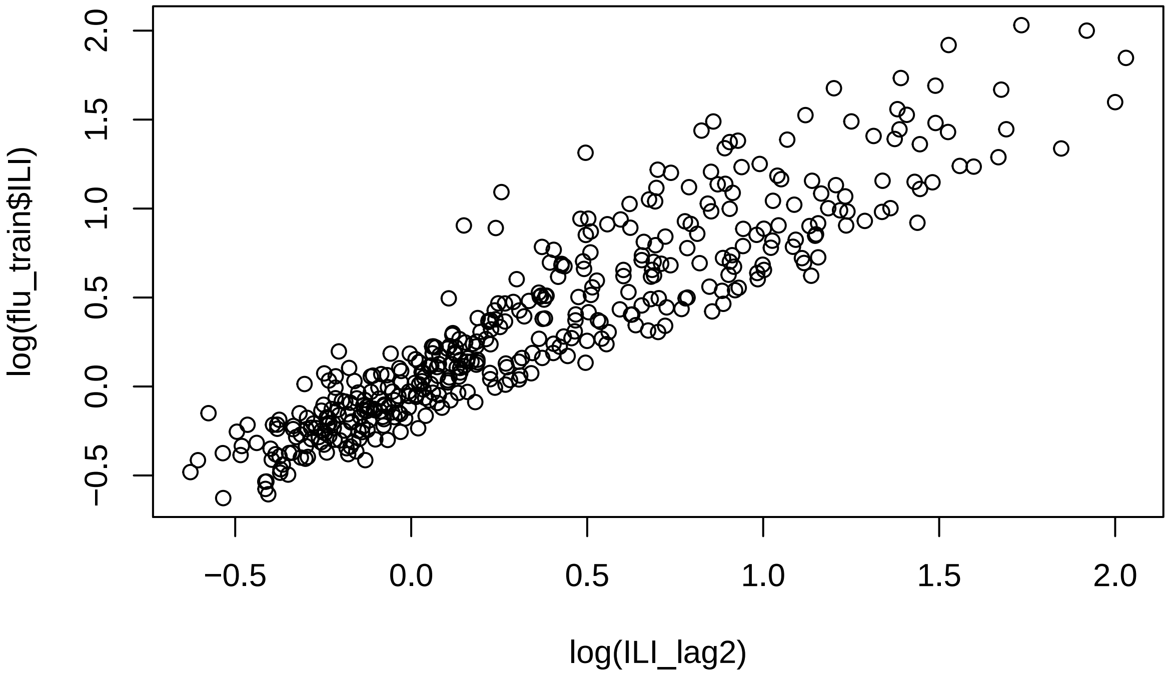
In these commands, **the value of -2 passed to lag means to return 2 observations before the current one**; a positive value would have returned future observations. The parameter **na.pad=TRUE means to add missing values for the first two weeks of our dataset, where we can't compute the data from 2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

How many values are missing in the new ILILag2 variable?

```
summary(ILI_lag2)
```

```
##      Index        ILI_lag2
##  Min.   :  1   Min.   :0.5341
##  1st Qu.:105   1st Qu.:0.9010
##  Median :209   Median :1.2519
##  Mean   :209   Mean   :1.6754
##  3rd Qu.:313   3rd Qu.:2.0580
##  Max.   :417   Max.   :7.6189
##                NA's   :2
```

```
plot(log(ILI_lag2), log(flu_train$ILI))
```



### 4.3) Training a Time Series Model

Train a linear regression model on the FluTrain dataset to predict the log of the ILI variable using the Queries variable as well as the log of the ILILag2 variable. Call this model FluTrend2.

Which coefficients are significant at the p=0.05 level in this regression model?

```
flu_trend_2 = lm(log(ILI) ~ Queries + log(ILI_lag2), data = flu_train)

summary(flu_trend_2)
```

```
##
## Call:
## lm(formula = log(ILI) ~ Queries + log(ILI_lag2), data = flu_train)
##
## 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(ILI_lag2)  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
```

### 4.4) Training a Time Series Model

**flu_trend_2 is a stronger model than flu_trend_1 on the training set**

### 5.1) Evaluating the Time Series Model in the Test Set

So far, we have only added the ILI_lag2 variable to the flu_train data frame. To make predictions with our flu_train_2 model, we will also need to add ILI_lag2 to the flu_test data frame (note that adding variables before splitting into a training and testing set can prevent this duplication of effort).

Modify the code from the previous subproblem to add an ILI_lag2 variable to the flu_test data frame. How many missing values are there in this new variable?

```
ILI_lag2 = stats::lag(zoo(flu_test$ILI), -2, na.pad = TRUE)

flu_test$ILI_lag2 = coredata(ILI_lag2)

summary(flu_test$ILI_lag2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.9018  1.1359  1.3409  1.5188  1.7606  3.6002       2
```

### 5.2) Evaluating the Time Series Model in the Test Set

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 flu_test was recorded one week after the last observation in flu_train. From this, we can identify how to fill in the missing values for the ILI_lag2 variable in flu_test.

Which value should be used to fill in the ILI_lag2 variable for the first observation in flu_test? * The ILI value of the second-to-last observation in the flu_train df Which value should be used to fill in the ILILag2 variable for the second observation in FluTest? * The ILI value of the last observation in the flu_train df

### 5.3) Evaluating the Time Series Model in the Test Set

Fill in the missing values for ILI_lag2 in flu_test. In terms of syntax, you could set the value of ILILag2 in row “x” of the flu_test data frame to the value of ILI in row “y” of the flu_train data frame with “FluTest$ILILag2[x] = FluTrain$ILI[y]”. Use the answer to the previous questions to determine the appropriate values of “x” and “y”. It may be helpful to check the total number of rows in FluTrain using str(FluTrain) or nrow(FluTrain).

```
flu_test$ILI_lag2[2] = flu_train$ILI[417]
flu_test$ILI_lag2[1] = flu_train$ILI[416]
head(flu_test)
```

```
##                      Week      ILI   Queries ILI_lag2
## 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
```

### 5.4) Evaluating the Time Series Model in the Test Set

Obtain test set predictions of the ILI variable from the flu_trend_2 model, again remembering to call the exp() function on the result of the predict() function to obtain predictions for ILI instead of log(ILI).

```
pred_test_2 = exp(predict(flu_trend_2, newdata = flu_test))

head(pred_test_2)
```

```
##        1        2        3        4        5        6 
## 2.482236 2.411829 2.140941 1.907817 1.971504 2.081855
```

What is the test-set RMSE of the flu_trend_2 model?

```
SSE_test_2 = sum((pred_test_2 - flu_test$ILI)^2)

RMSE_2 = sqrt(SSE_test_2 / nrow(flu_test))

RMSE_2
```

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

### 5.5) Evaluating the Time Series Model in the Test Set

Which model obtained the best test-set RMSE

```
pred_test = exp(predict(flu_trend_1, newdata = flu_test))

SSE_test_1 = sum((pred_test - flu_test$ILI)^2)

RMSE_1 = sqrt(SSE_test_1 / nrow(flu_test))

RMSE_1
```

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