Big Data Analysis Using Machine Learning for Social Scientists and Criminologists
Big Data Analysis Using Machine Learning for Social Scientists and Criminologists

By
Juyoung Song and Tae Min Song
# TABLE OF CONTENTS

Installation and Use of R ................................................................. 1  
   Installation of R ........................................................................... 1  
   Use of R ...................................................................................... 7  

Scientific Research Design .............................................................. 35  
   Research Concepts ...................................................................... 36  
   Variable Measurement ................................................................. 37  
   Unit of Analysis .......................................................................... 39  
   Sampling and Hypothesis Testing ................................................. 39  
   Statistical Analysis ..................................................................... 44  

Overview of Machine Learning ...................................................... 118  
   Introduction ................................................................................. 118  
   Machine Learning Training Data ............................................... 122  

Development of a Cyber bullying Prediction Model Based on Machine Learning ............................................. 124  
   Naïve Bayes Classification Model ................................................. 124  
   Logistic Regression Model ............................................................ 130  
   Random Forest Model ................................................................. 134  
   Decision Tree Model .................................................................. 141  
   Neural Network Model ............................................................... 149  
   Support Vector Machine Model ................................................. 162  
   Association Analysis .................................................................. 170  
   Cluster Analysis and Segmentation ............................................. 179  

Machine Learning Model Evaluation .......................................... 186  
   Machine Learning Model Evaluation Using  
   Misclassification Tables ............................................................... 189  
   Machine Learning Model Evaluation Using ROC Curves ............ 208  

Artificial Intelligence ..................................................................... 215  
   Calculate the Effect of Input Variables on Output Variables  
      (Prediction Probability) ............................................................. 215
Using Training Data with Input Variables to Create Dependent Variables ................................................................. 221
Creating Data with the Same Training-Data and Predicted-Data Classifications .......................................................... 225
Evaluating Existing Training Data and High Quality Training Data ............................................................................. 228
Creating an Artificial Intelligence with Machine Learning .............................................................................. 230
Visualization ................................................................................................................................................................. 236
Visualization of Text Data ........................................................................................................................................... 236
Visualization of Time Series Data ............................................................................................................................... 239
Visualization of Geographical Data .......................................................................................................................... 250
Developing Machine Learning–Based Predictive Models of Adverse Drug Responses ................................................. 258
Introduction ................................................................................................................................................................. 258
Research Subjects and Analysis Method ..................................................................................................................... 263
Result ............................................................................................................................................................................ 269
Discussion and Conclusion ......................................................................................................................................... 302
Index ................................................................................................................................................................................ 307
The R program (simply “R” hereafter) is an open-source (i.e., the source code is made public so that anyone can use, modify, or redistribute the code for free) program developed for statistical analysis and visualization. It is an object-oriented language based on objects that can take scripts used in one analysis and reuse them in another analysis. R is an open-source language derived from the S language developed at Bell Laboratories in 1976. In 1995, the source was made public by Robert Gentleman and Ross Ihaka at the University of Auckland in New Zealand. Since then, it has been continuously improved by the R core development team. It is executed in interactive mode so the execution results can be seen quickly. R is an object-oriented language that can reuse the instructions (i.e., scripts) used in analysis for other analyses. R is useful for developing functions and packages, which are collections of scripts that perform specific functions, and is widely used among statisticians for statistical software development and data analysis. Today, packages and functions developed by several experts are publicly available on CRAN (Comprehensive R Archive Network), and the program’s usefulness is continuously increasing.

Installation of R

Anyone can install and use R by downloading the program from the R project homepage (http://www.r-project.org). To use R for graphing or visualization, the Java program for modern Windows operating systems (32 bit or 64 bit) must be installed. The processes for installing R and Java follow.

1. Download R-3.5.0-win.exe from the R project homepage and run the program.
Installation and use of R

② Set the installation language to English and click [OK].

Select Setup Language

Select the language to use during the installation:

- English

[OK] [Cancel]

③ Click [Next] to begin the installation. As information about the installation appears, continue to click [Next].

④ Select the location where the R program will be installed. If using the default folder, click [Next].
⑤ Install the components that are appropriate for operating on the PC where the program will be installed, and click [Next].

⑥ In the startup options, select “No (accept defaults)” and click [Next].

⑦ Select the R program’s start menu folder and click [Next].

⑧ Select the additional installation items (use defaults) and click [Next].
The installation-progress screen will appear. When the “installation complete” screen appears, click [Finish].

Search for the Java program (JDK SE development) on Google. On the download page, download the JDK file suitable for your PC and run jdk-8u40-windows-x64.
When the Java installation screen appears, click [Next]. Select the installation components (select the default items) and click [Next].

When the Java program’s installation is complete, click [Close] to finish the Java installation.
After the installation is complete, go to the Windows start menu and click [All Programs] → [R] or click the R icon installed on the desktop. When closing the program, click the ‘X’ on the window or enter ‘q()’.

★ Changing the R-Console to English

① Open a text editor, e.g., Wordpad, with administrator privileges and open the following file.
① Modify the content of the file as shown below.

-   (text above)
-   ## Language for messages
-   language = en
-   (text below)

③ Run R-3.5.0 again and the English R-Console will appear.

**Use of R**

R is a script-command–based program. Packages needed for various analyses can be installed and used as libraries.

1) Installing and Loading Packages

R is open source so it has no distribution limitations; i.e., R can be used to create and provide new solutions, even if they will be sold commercially. R can install and load a variety of packages, depending on the analysis method (statistical analysis, machine learning, visualization, etc.). Packages can be freely downloaded from the CRAN site (www.r-project.org) and installed. R comes with several basic packages, and 12,000 additional packages are available on CRAN (12,087 packages registered as of February 5, 2018). An internet connection is necessary when first installing additional packages. Packages can be installed from the homepage’s CRAN mirrors by using the install.packages() function or the “Install Packages” command on the menu bar in R. The mirror site has copies of the same content at multiple locations to prevent a large amount of traffic congregating at a single site. As of May 10, 2018, 161 mirror sites were operating in 48 regions, including the ‘0-Cloud’. The United States has 15 mirror sites that can be used.

Script Example: Create a word cloud of keywords for strain and delinquency factors in cyber bullying.
> setwd("c:/cyberbullying_methodology"): Set the working directory
> install.packages(’wordcloud’)  
- Install the package that processes word clouds.
> library(wordcloud) : Load the package that processes word clouds.
> freq=c(2269,1338,3515,7269,5844,1101,32816,1503,32084,5849,8949,2348,858,539,617,1085,6452,784,1852,1764,2496,29473,24413,488,799,2253,1497,1153)  
- Assign keyword frequencies for cyber bullying strain and delinquency factors to the freq vector.
> library(RColorBrewer) : Load the package for displaying color.
> palette=brewer.pal(9,”Set1”)  
- Assign RColorBrewer’s nine text colors to a palette variable.
> wordcloud(key,freq,scale=c(4,1),rot.per=.12,min.freq=100, random.order=F, random.color=T,colors=palette) : Display the word cloud.
> savePlot("cyber_bullying_strain_wordcloud",type=”png”)  
- Save the results as an image file.
# Installation and use of R

```r
# machine learning wordcloud 2018. 5. 10.

setwd("c:/cyberbullying_methodology")
install.packages("wordcloud")

Warning: package "wordcloud" is in use and will not be installed

library(wordcloud)

      "Class-society", "Korea", "Female_dislike", "Interested_soldier", "Traffic-accidents",
      "Personal-broadcasting")

freq=c(1289, 1308, 3915, 7249, 5848, 1301, 30216, 1509, 52064, 9949, 2365, 858, 559, 617, 1195,
      6452, 7658, 1892, 1764, 2496, 5947, 24113, 489, 739, 2253, 1497, 1159)

library(RColorBrewer)
pal<-brewer.pal(9,"Set1")
wordcloud(key,freq, scale=c(4,1),rot.per=10,min.freq=100,random.order=0,
          random.color=0,colors=pal)
saveplot("cyber_bullying_straws_wordcloud", type="pdf")
```

2) Value Assignment and Calculation

1. Run the R shortcut on the Windows desktop. In the initial screen, enter a script in the column after the prompt ‘>’. Press the ‘ENTER’ key to run it.

2. In R, saving the execution results (values) to objects or variables is called assigning. Use “=” or “<-” to assign values in R. (This book uses ‘=’.)

3. When an R script is long, use “;” to connect the next line.

4. Use ‘;’ to connect several scripts.

5. R has the following rules for using variables.
   - Variable names are case sensitive.
   - Variable names can use English letters, numbers, periods (.), and underscores (_); however, the first character cannot be a number or underscore. When numbers are used as variables, “X” is automatically added to the first character.
   - The reserved words in the R system (if, else, NULL, NA, in, etc.) cannot be used as variable names.

6. Functions are collections of scripts that take an argument-type value as input and return the calculated result value. In R, functions can be used to make programs more concise.

7. The following operators are available: Operators [+ , - , *, / , % (modulus), ^ (exponent), etc.] and R’s internal functions [sin(), exp(), log(), sqrt(), mean(), etc.].

- Saving formulas using operators

> pie=3.1415 : Assign 3.1415 to pie.
> x=100 : Assign 100 to x.
> y=2*pie+x : Assign 2 × pie + x to y.
> y : Display the value of y on the screen.

- Saving formulas using internal functions

> x=c(75, 80, 73, 65, 75, 83, 73, 82, 75, 72) : Assign 10 vector values (weights) to x.
> mean(x) : Display the mean of x on the screen.
> sd(x) : Display the standard deviation of x on the screen.
Installation and use of R

1 To repeat a previously performed task, press the up-arrow key.
2 To end the R program, click the ‘X’ on the window or enter ‘q()’.

3) Basic Data Types in R

1) Set the directory where all of the objects (functions, data, etc.) that are used in R will be saved
> setwd("c:/cyberbullying_methodology")

2) The basic data types in R are as follows.
   - Numeric type: Uses arithmetic operators [+,-,*,/,%% (modulus), ^ (exponent), etc.] to calculate results.
     > x = sqrt(50*(100^2))

   - Character type: Groups together strings of text with single quotes (‘’) or double quotes (“”).
     > v_name = 'machine learning modeling'

   - NA type: Use this when a value is not determined.
     > x = mean(c(75, 80, 73, 65, 75, 83, 73, 82, 75, NA))

   - Factor type: Use this to convert character-type data into numeric type.
     > x = c("a", "b", "c", "d"); x_f = factor(x)
Date and time format: Use this when analyzing a certain time period or time.

```r
> x = (as.Date('2018-10-07') - as.Date('2017-10-06'))
```

4) Data Structures in R

In R, data is managed through vector, matrix, array, and list-type data structures.

(1) Vector

A vector is the basic data structure in R. A vector is a data object that combines and stores several data items. Vector in R use the `c()` function to assign values.

```r
> x=c(75, 80, 73, 65, 75, 83, 73, 82, 75, 72)
- Assign the weights of ten people to variable x as a vector.
> y=c(5, 2, 3, 2, 5, 3, 2, 5, 7, 4)
- Assign the weight loss of 10 people to variable y as a vector.
> d=x - y
- Subtract vector y from vector x and assign the result to vector d.
> d : Display the value of vector d on the screen.
- Subtract the value of the 4th element in vector x (65) from the value of the 4th element in vector y (2) and assign the result to variable e.
> e : Display the value of variable e on the screen.
```
Character-type data management

```r
> x=c('flaming','harassment','denigration','impersonation','outing',
' exclusion','cyberstalking') : Assign character data to vector x.
```

Use the seq() function or `:` to assign sequential data to a vector.

```r
> x=seq(10, 150, 10)
> x=30:50
```

- Create a sequence of numbers from 10 to 150 with an increment of 10, and assign the results to vector x.
- Create a sequence of numbers from 30 to 50 with an increment of 1 and assign the results to vector x.
(2) Matrix

A matrix is two-dimensional vector data structures that additionally have rows and columns. The matrix() function is used for data management.

```r
> x_matrix=matrix(c(75, 80, 73, 65, 75, 83, 73, 82, 75, 72, 77, 76),
                  nrow=4, ncol=3)
```
- Create a matrix with 4 rows and 3 columns containing 12 people’s weights, and assign the results to x_matrix.

```r
> x_matrix
```
- Display the value of x_matrix on the screen.

```r
> x_matrix[2,1]
```
- Display the value of the element at row 2, column 1 of x_matrix on the screen.

(3) Array

An array is three or more dimensions and can extend matrices multidimensionally. The array() function is used for data management.

```r
> x=c(75, 80, 73, 65, 75, 83, 73, 82, 75, 72, 77, 76)
```
- Assign the weights of 12 people to vector x.

```r
> x_array=array(x, dim=c(3, 3, 3))
```
- Assign vector x to the x_array variable as a 3D structure.

```r
> x_array
```
- Display the value of the array variable x_array on the screen.

```r
> x_array[2,2,1]
```
- Display the value of the element at [2, 2, 1] in x_array on the screen.
List

A list is a type of matrix or array that can specify the data type in the format of (address, value).

```r
> x_address <- list(name = "Pennsylvania State University Schuylkill, Criminal Justice", address = "200 University Drive, Schuylkill Haven, PA 17972", homepage = "http://www.sl.psu.edu")
- Assign the address to the list format variable `x_address`.
> x_address
```

```r
> x_address <- list(name = "Sahmyook university, department of health management", address = "815, Hwarang-ro, Nowon-gu, Seoul, 01795, KOREA", homepage = "https://www.syu.ac.kr")
> x_address
```

(4) List
5) Using Functions in R

Users can use R’s built-in functions or they can use function( ) to create their own functions. User-defined functions should use the following basic format.

```
Function name = function(argument1, argument2, ...) {
  Calculation formula or executable program
  return(calculation results or return value)
}
```

Exercise 1: Find the sample size using the confidence level and sampling error
- Formula: \( n = \left( \pm Z \right)^2 \times P(1 - P)(SE)^2 \)
- Create a function (SZ) to find the sample size when a phone survey is conducted to analyze the state of school bullying with a sample error of 3% at a confidence level of 95% (Z = 1.96) at \( p = 0.5 \).
Exercise 2: Find the standard score
- The standard score is a measure of the extent to which an observed value differs from the mean. It can be used to find the data’s relative position. The sum of the observed values’ standard scores is 0.
- Formula: \( Z_i = \frac{(X_i - \bar{X})}{S_X} \)
- Create a function (ZC) to find the standard score after measuring the weights of 10 people.

Exercise 3: Find the population variance

```r
> setwd("c:/cyberbullying_methodology")
- Set the working directory.
> cyber_bullying=read.table(file="cyber_bullying_descriptive_analysis.txt",header=T)
- Import the text data and assign it to cyber_bullying.
> attach(cyber_bullying)
- Attach cyber_bullying as execution data.
> VAR=function(x) var(x)*(length(x)-1)/length(x)
- Create a new function with the format of “function (argument or input value) formula”.
- length(x): Calculate the sample size of the x variable that is passed to the VAR function as an argument.
- Create the function (VAR) to find the population variance for the x argument.
> VAR(Onespread)
- Call the VAR function and calculate the population variance for Onespread.
> sqrt(VAR(Onespread))
- Call the VAR function and calculate the population standard deviation for Onespread.
> VAR(Twospread)
- Call the VAR function and calculate the population variance for Twospread.
> sqrt(VAR(Twospread))
- Call the VAR function and calculate the
population standard deviation for Twospread.

6) Basic Program in R (Conditional Statements and Loop Statements)

- R provides conditional statements, which determine the execution flow, as well as loop statements, which repeat the same statement multiple times.
- Conditional statements use comparative operators [equal (==), not equal (!=), greater than or equal (>=), greater than (>), less than or equal (<=), and less than (<)].
- Conditional statements are formatted as follows.

```r
if(condition formula) {
  <Calculation to be performed if condition is true>
}
else {
  <Calculation to be performed if condition is false>
}
```

Exercise 4: Using conditional statements

- Create a function (F) that returns the mean of vector x, which holds the weights of 10 people if argument is ’1’, and which returns the standard deviation if argument is not ’1’.
- The format for loop statements is as follows.
- The “number of times” used in a for loop is either the “vector data” or “n: number of times.”

```r
for(loop variable in number of times) {
    Executed statement
}
```

Exercise 5: Using loop statements
- Create a function (F) that finds the sum of numbers from 1 to a given number.

```r
# R Console
> # # iteration structure
> F=function(a){
>     result=0
>     for(i in 1:a){
>         result=result+i
>     }
>     return(result)
> }
> F(100)
[1] 5050
> F(500000)
[1] 12500250000
> F(2018)
[1] 2037171
```
7) Using Variables in R Data Frames

Variables can be used for statistical analysis in R as follows.

(1) Using “data$variables”

```r
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_analysis.txt",header=T)
> cyber_bullying_1=read.table(file="cyber_bullying_descriptive_analysis_1.txt",header=T)
> sd(cyber_bullying$Onespread)/mean(cyber_bullying$Onespread)
```

- Use the Onespread variable of the cyber_bullying data frame to find the variation coefficient.

(2) Using the attach(data) function

```r
> attach(cyber_bullying)
- The attach function attaches execution data as a “data” argument.
> sd(Onespread)/mean(Onespread)
- Unlike “data$variable”, after attach is executed, the variable alone can be used to find the variation coefficient.
```

(3) Using the with(data, script) function

```r
> with(cyber_bullying_1,sd(Onespread)/mean(Onespread))
- A script can be executed using the data frame variable via the with( ) function without using the attach function .
```
8) Creating R Data Frames

R can create data frames with a variety of formats. The most often used data frame is the two-dimensional matrix, which has rows and columns. Data frames are also called data sets. Their columns are called variables, and their rows are called records.

(1) Creating data frames from vectors

- Use the `data.frame()` function.

```r
> V0=1:10 : Assign numbers 1–10 to the V0 vector.
> V1=c(4, 7, 16, 12, 8, 11, 14, 9, 4, 8) : Assign 10 numbers to the V1 vector, and the next six vectors.
```
> V2=c(3, 5, 11, 11, 6, 6, 13, 4, 3, 7)
> V3=c(2, 5, 12, 17, 9, 6, 15, 6, 3, 9)
> V4=c(1, 0, 14, 12, 0, 0, 4, 1, 0, 1)
> V5=c(3, 2, 15, 8, 2, 6, 2, 1, 4)
> V6=c(6, 3, 12, 10, 3, 4, 5, 6, 3, 5)
> V7=c(3, 2, 19, 15, 7, 8, 14, 4, 2, 8)
> willard_data=data.frame(ID=V0,flaming=V1,harassment=V2,
> denigration=V3,impersonation=V4,outing=V5,exclusion=V6,
> cyberstalking=V7)

- Assign the 8 vectors (V0–V7) to the willard_data data frame object.

> willard_data

- Display the value of the willard_data data frame on the screen.

(2) Creating data frames from text files

- Use the read.table() function.

> setwd("c:/cyberbullying_methodology") : Set the working directory.
> willard_data=read.table(file="willard_data.txt",header=T)

- Assign the “willard_data.txt” file to the willard_data object via the data frame.

> willard_data : Display the value of the willard_data object on the screen.
(3) Creating data frames from CSV files

- Use the read.table() function.

```
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.csv",header=T)
> willard_data
```

(4) Creating data frames from SPSS files

- Use the read.spss() function.

```>
> install.packages('foreign')
```

- Install a package for reading external data created by statistics software other than R; e.g., SPSS or SAS.