

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

Cambridge
Scholars
Publishing



Big Data Analysis Using Machine Learning for Social Scientists
and Criminologists

By Juyoung Song and Tae Min Song

This book first published 2019

Cambridge Scholars Publishing

Lady Stephenson Library, Newcastle upon Tyne, NE6 2PA, UK

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

Copyright © 2019 by Juyoung Song and Tae Min Song

All rights for this book reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owner.

ISBN (10): 1-5275-3388-3

ISBN (13): 978-1-5275-3388-2

This work was supported by the Ministry of Education of the
Republic of Korea and the National Research Foundation of Korea
(NRF-2016S1A5A2A03925702)

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

INSTALLATION AND USE OF R

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.



If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe file. Both graphical and command line versions are available.

[Frequently asked questions](#)

- Does R run under my version of Windows?
- How do I update packages in my previous version of R?
- Should I run 32-bit or 64-bit R?

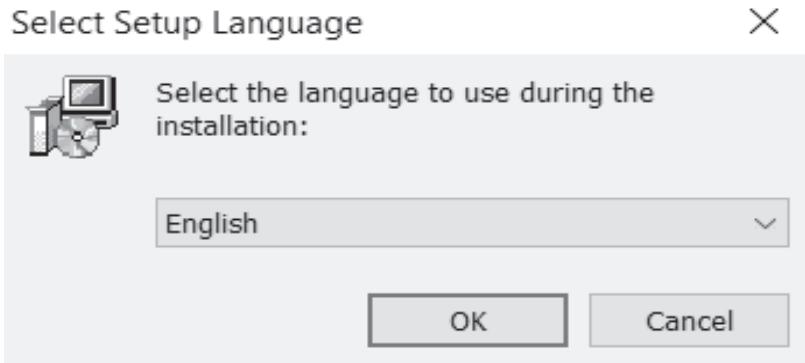
Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

[Other builds](#)

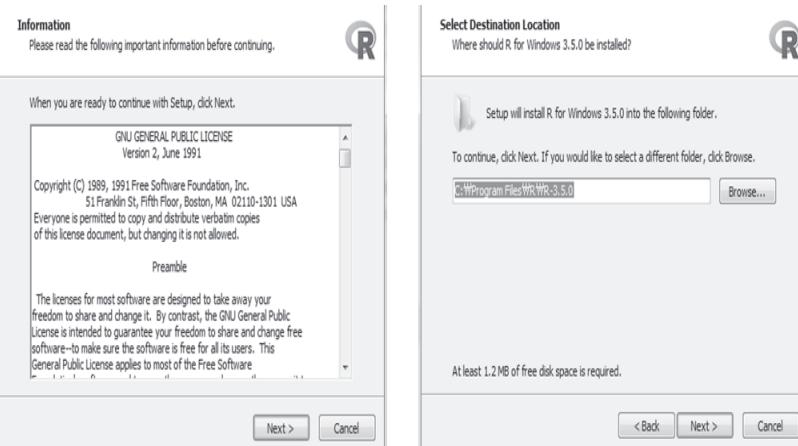
- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is
`<CRAN MIRROR>/bin/windows/base/release.htm`

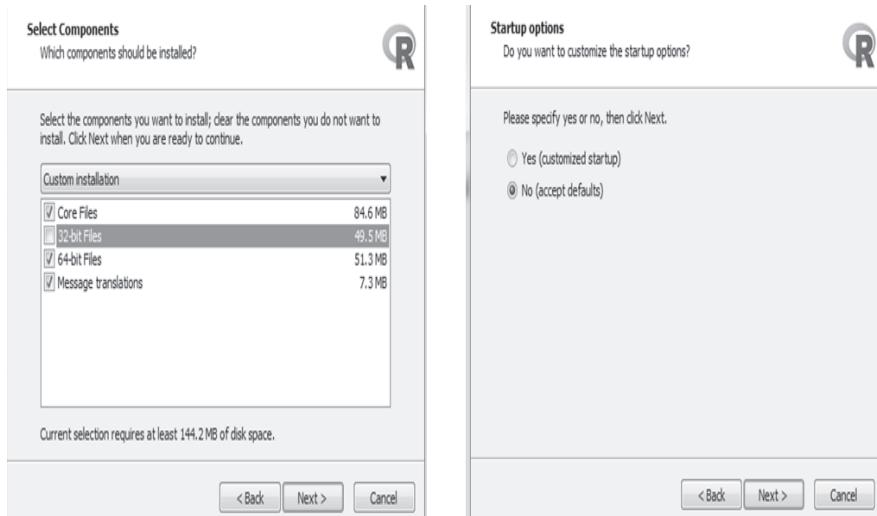
- ② Set the installation language to English and click [OK].



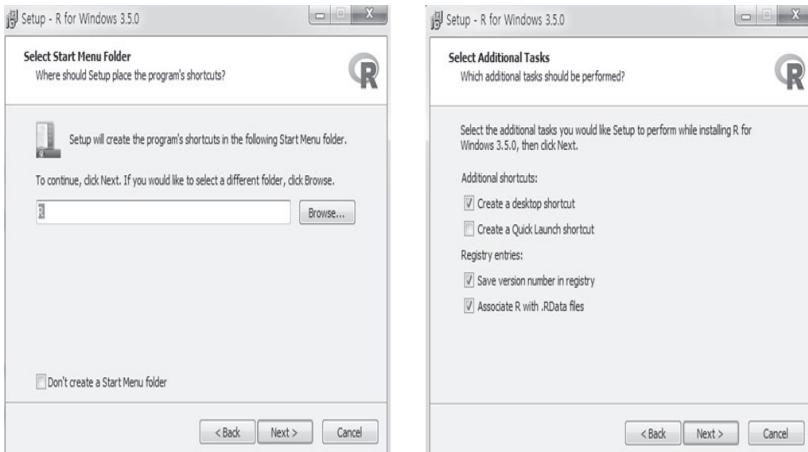
- ③ Click [Next] to begin the installation. As information about the installation appears, continue to click [Next].
- ④ Select the location where the R program will be installed. If using the default folder, click [Next].



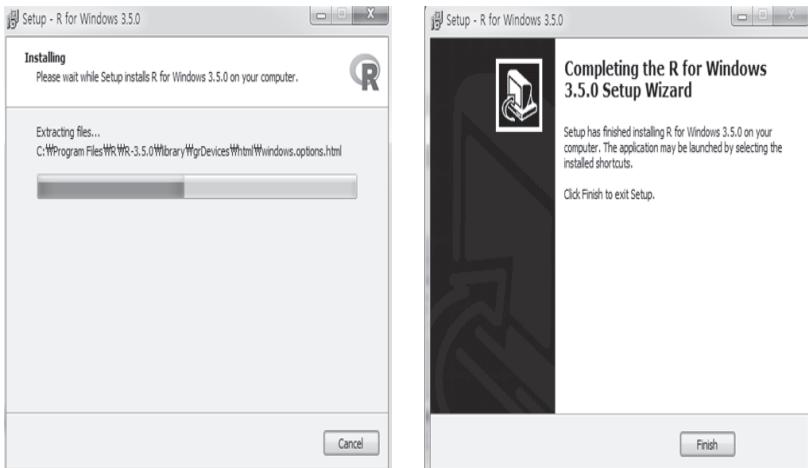
- ⑤ Install the components that are appropriate for operating on the PC where the program will be installed, and click [Next].
- ⑥ In the startup options, select “No (accept defaults)” and click [Next].



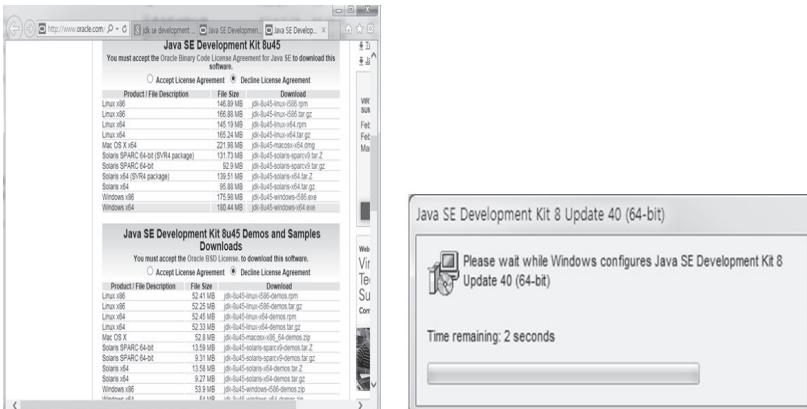
- ⑦ Select the R program’s start menu folder and click [Next].
- ⑧ Select the additional installation items (use defaults) and click [Next].



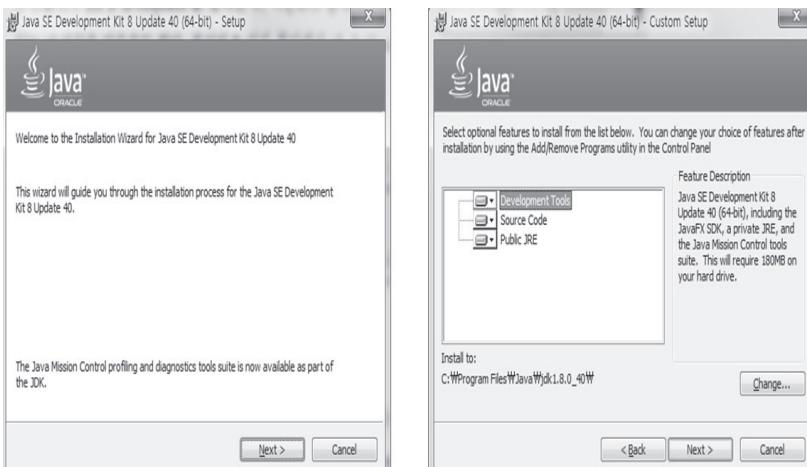
- ⑨ The installation-progress screen will appear. When the “installation complete” screen appears, click [Finish].



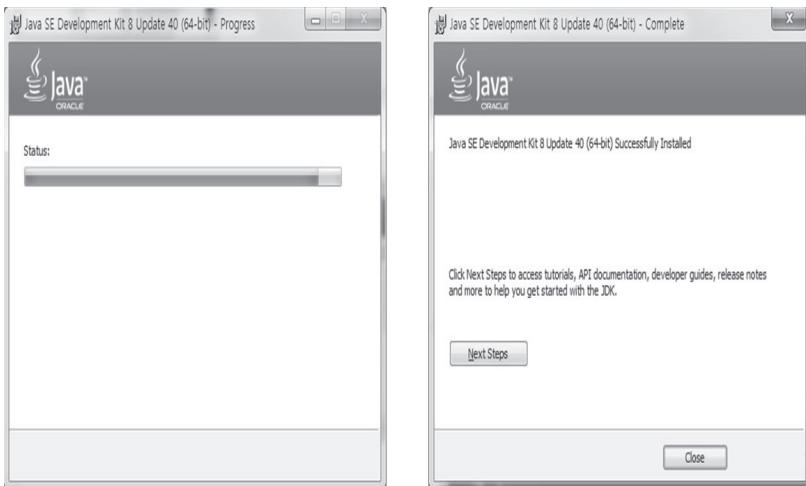
- ⑩ Search for the Java program (JDK SE development) on Google. On the download page, download the JDK file suitable for your PC and run jdk-8u40-windows-x64.



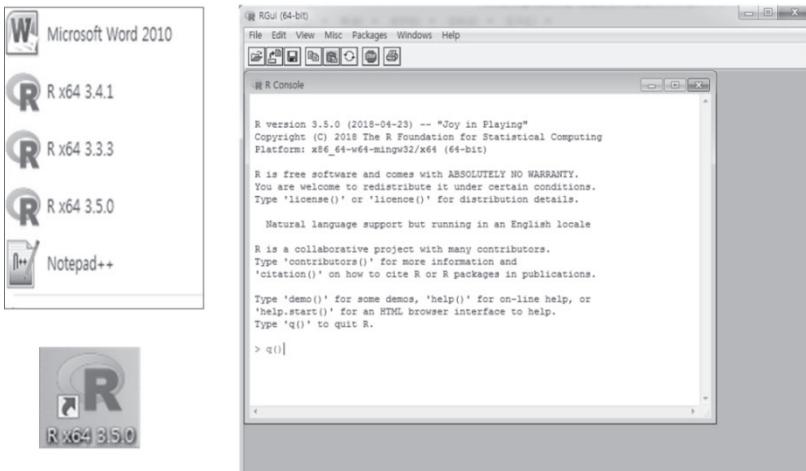
- ⑪ When the Java installation screen appears, click [Next]. Select the installation components (select the default items) and click [Next].



- ⑫ When the Java program's installation is complete, click [Close] to finish the Java installation.



- ⑬ 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.

- C:\Program Files\R\R-3.5.0\etc\Rconsole

② 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

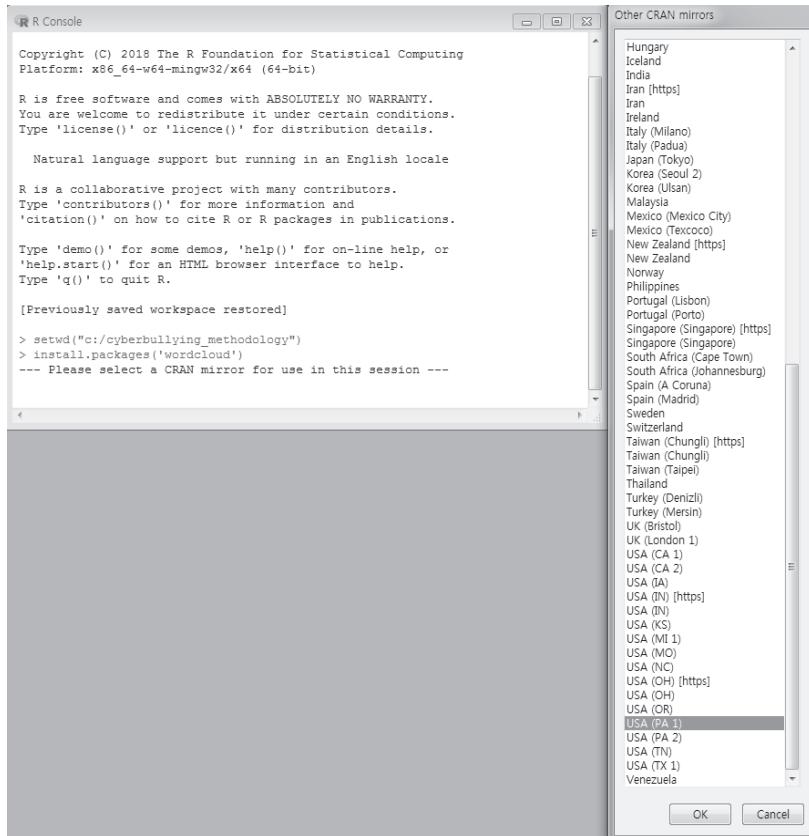
R is a script-command-based program. Packages needed for various analyses can be installed and used as libraries.

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.

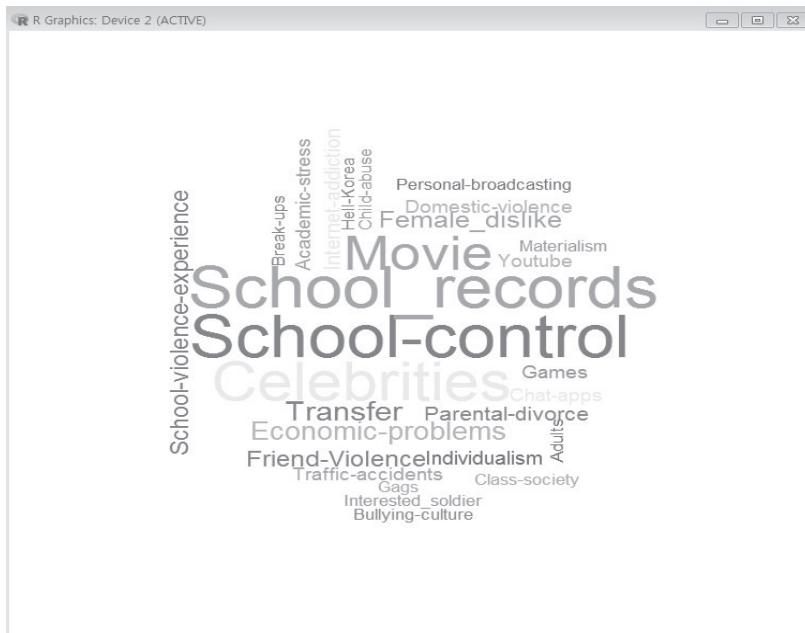
Script Example: Create a word cloud of keywords for strain and delinquency factors in cyber bullying.

```
> 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.
> key=c('Domestic-violence','Child-abuse','Parental-divorce','Economic-
    problems','Friend-Violence','Break-ups','School-control','Academic-
    stress','School_records','School-violence-experience','Transfer',
    'Individualism','Materialism','Bullying-culture','Class-society','Hell-
    Korea','Female_dislike','Interested_soldier','Traffic-accidents','Games',
    'Internet-addiction','Celebrities','Movie','Adults','Gags','Chat-apps',
    'Youtube','Personal-broadcasting') : Assign keywords for cyber bullying
    strain and delinquency factors to the key vector.
> 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.
> palete=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=palete) : Display the word cloud.
> savePlot("cyber_bullying_strain_wordcloud",type="png")
    - Save the results as an image file.
```



```
R Console

> ## 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)
>
> key=c('Domestic-violence','Child-abuse','Parental-divorce','Economic-problems',
+ 'Friend-Violence','Break-ups','School-control','Academic-stress','School_records',
+ 'School-violence-experience','Transfer','Individualism','Materialism','Bullying-culture',
+ 'Class-society','Hell-Korea','Female_dislike','Interested_soldier','Traffic-accidents',
+ 'Games','Internet-addiction','Celebrities','Movie','Adults','Gags','Chat-apps','Youtube',
+ 'Personal-broadcasting')
>
> 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)
> library(RColorBrewer)
> palette=brewer.pal(9,"Set1")
> wordcloud(key,freq,scale=c(4,1),rot.per=.12,min.freq=100,random.order=F,
+ random.color=T,colors=palette)
> savePlot("cyber_bullying_strain_wordcloud",type="png")
> |
```



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.
```



The screenshot shows an R console window titled "R Console". The window contains the following R code and its execution results:

```

> ## value assignment
>
> setwd("c:/cyberbullying_methodology")
>
> pie=3.1415
> x=100
> y=2*pie+x
> y
[1] 106.283
>
> ## internal function
>
> x=c(75,80,73,65,75,83,73,82,75,72)
> mean(x)
[1] 75.3
> sd(x)
[1] 5.313505
>

```

- ⑧ To repeat a previously performed task, press the up-arrow key.
- ⑨ 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 set.

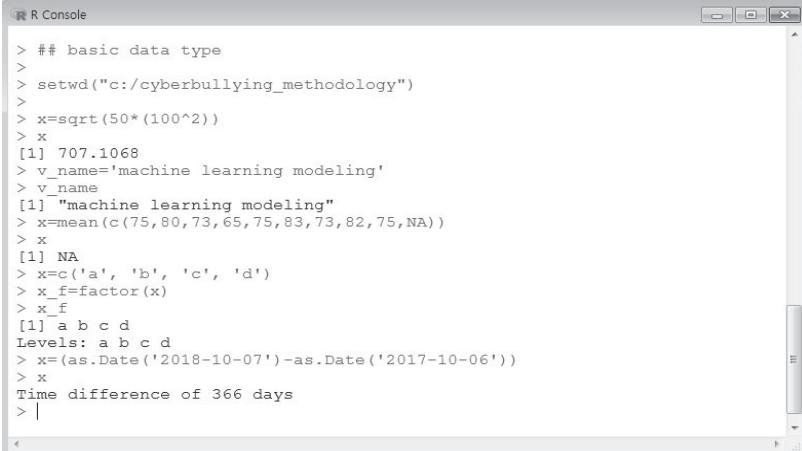
```
> x = mean(c(75, 80, 73, 65, 75, 83, 73, 82, 75, NA))
```

- 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.

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



The screenshot shows the R Console window with the following session history:

```
R Console
> ## basic data type
>
> setwd("c:/cyberbullying_methodology")
>
> x=sqrt(50*(100^2))
> x
[1] 707.1068
> v_name='machine learning modeling'
> v_name
[1] "machine learning modeling"
> x=mean(c(75,80,73,65,75,83,73,82,75,NA) )
> x
[1] NA
> x=c('a', 'b', 'c', 'd')
> x_f=factor(x)
> x_f
[1] a b c d
Levels: a b c d
> x=(as.Date('2018-10-07')-as.Date('2017-10-06'))
> x
Time difference of 366 days
> |
```

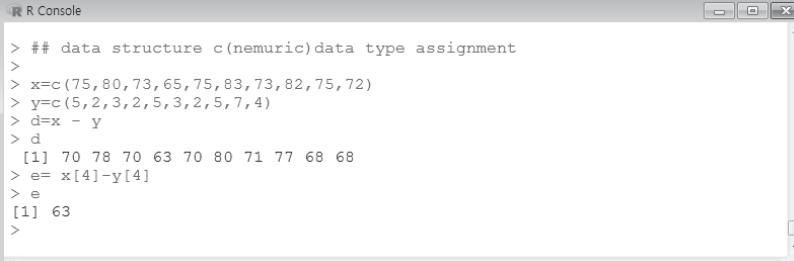
4) Data Structures in R

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.

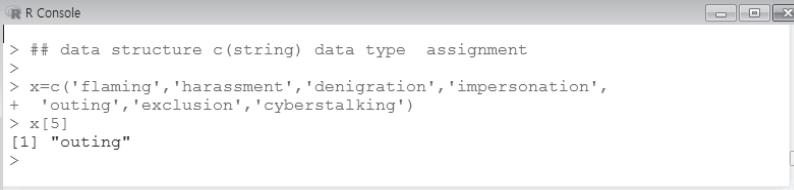
```
> x=c(75, 80, 73, 65, 75, 83, 73, 82, 75, 72)
  - Assign the weights of ten people to variable x as a vector.
> y=c(5, 2, 3, 2, 5, 3, 2, 5, 7, 4)
  - Assign the weight loss of 10 people to variable y as a vector.
> d=x - y
  - Subtract vector y from vector x and assign the result to vector d.
> d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



```
> ## data structure c(numeric) data type assignment
>
> x=c(75,80,73,65,75,83,73,82,75,72)
> y=c(5,2,3,2,5,3,2,5,7,4)
> d=x - y
> d
[1] 70 78 70 63 70 80 71 77 68 68
> e= x[4]-y[4]
> e
[1] 63
>
```

- Character-type data management

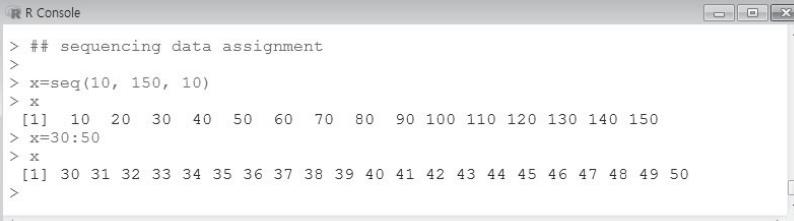
> x=c('flaming','harassment','denigration','impersonation','outing','exclusion','cyberstalking') : Assign character data to vector x.
 > x[5] : Display the fifth element value of vector x on the screen.



```
> ## data structure c(string) data type assignment
>
> x=c('flaming','harassment','denigration','impersonation',
+ 'outing','exclusion','cyberstalking')
> x[5]
[1] "outing"
>
```

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

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



```
> ## sequencing data assignment
>
> x=seq(10, 150, 10)
> x
[1] 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
> x=30:50
> x
[1] 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
>
```

(2) Matrix

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

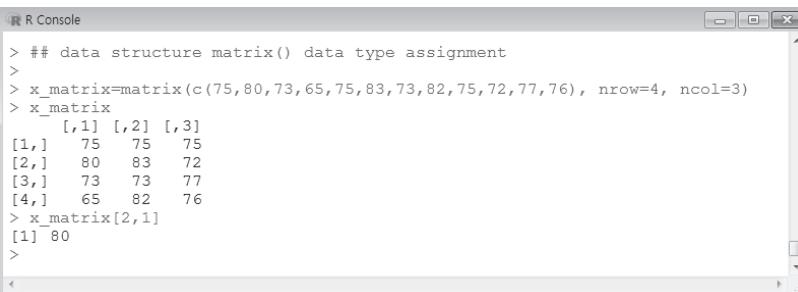
```
> x_matrix=matrix(c(75, 80, 73, 65, 75, 83, 73, 82, 75, 72, 77, 76),  
nrow=4, ncol=3)
```

- Create a matrix with 4 rows and 3 columns containing 12 people's weights, and assign the results to `x_matrix`.

```
> x_matrix : Display the value of x_matrix on the screen.
```

```
> x_matrix[2,1]
```

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



The screenshot shows the R console window. The code entered is:

```
> ## data structure matrix() data type assignment
>
> x_matrix=matrix(c(75, 80, 73, 65, 75, 83, 73, 82, 75, 72, 77, 76), nrow=4, ncol=3)
> x_matrix
     [,1] [,2] [,3]
[1,]    75    75    75
[2,]    80    83    72
[3,]    73    73    77
[4,]    65    82    76
> x_matrix[2,1]
[1] 80
>
```

(3) Array

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

```
> x=c(75, 80, 73, 65, 75, 83, 73, 82, 75, 72, 77, 76)
```

- Assign the weights of 12 people to vector `x`.

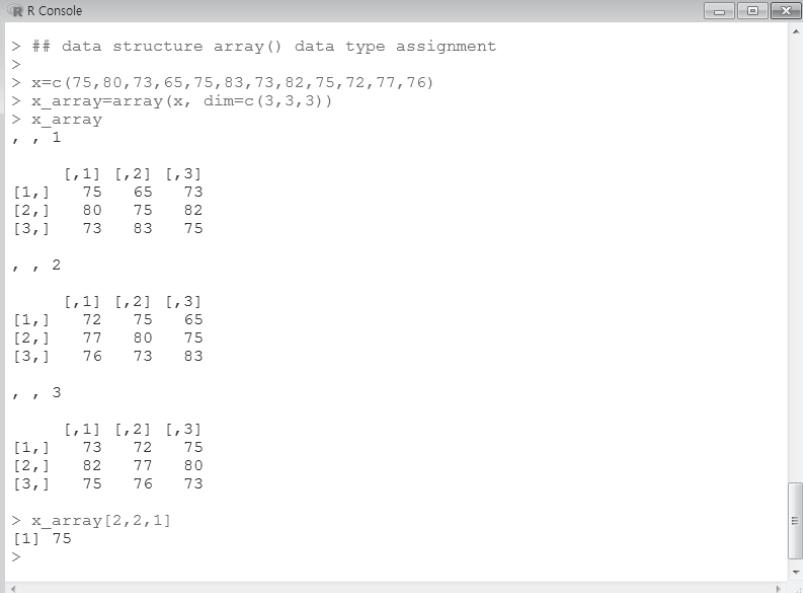
```
> x_array=array(x, dim=c(3, 3, 3))
```

- Assign vector `x` to the `x_array` variable as a 3D structure.

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

```
> x_array[2,2,1]
```

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



The screenshot shows an R console window with the title 'R Console'. The window contains the following R code and its output:

```

> ## data structure array() data type assignment
>
> x=c(75,80,73,65,75,83,73,82,75,72,77,76)
> x_array=array(x, dim=c(3,3,3))
> x_array
, , 1

 [,1] [,2] [,3]
[1,] 75   65   73
[2,] 80   75   82
[3,] 73   83   75

, , 2

 [,1] [,2] [,3]
[1,] 72   75   65
[2,] 77   80   75
[3,] 76   73   83

, , 3

 [,1] [,2] [,3]
[1,] 73   72   75
[2,] 82   77   80
[3,] 75   76   73

> x_array[2,2,1]
[1] 75
>

```

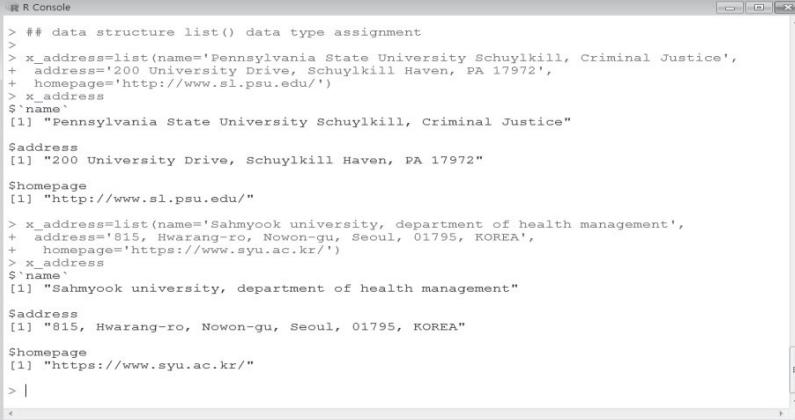
(4) List

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

```

> 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 : Display the value of the x_address variable on the screen.
> 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

```



```

R Console
> ## data structure list() data type assignment
>
> x_address=list(name='Pennsylvania State University Schuylkill, Criminal Justice',
+ address='200 University Drive, Schuylkill Haven, PA 17972',
+ homepage='http://www.s1.psu.edu/')
> x_address
$`name`
[1] "Pennsylvania State University Schuylkill, Criminal Justice"

$address
[1] "200 University Drive, Schuylkill Haven, PA 17972"

$homepage
[1] "http://www.s1.psu.edu/"

> 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
$`name`
[1] "Sahmyook university, department of health management"

$address
[1] "815, Hwarang-ro, Nowon-gu, Seoul, 01795, KOREA"

$homepage
[1] "https://www.syu.ac.kr/"
> |

```

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: $n = (\pm Z)^2 \times P(1 - P)/(SE)^2$
 - 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 $p = .5$.



```

R Console
> ## Function usage
> ## sample size
>
> SZ=function(p, z, s) {
+ n=z^2*p*(1-p)/s^2
+ return(n)
+ }
> SZ(0.5, 1.96, 0.03)
[1] 1067.111
>

```

● 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: $Z_i = (X_i - \bar{x})/s_x$
- Create a function (ZC) to find the standard score after measuring the weights of 10 people.

```

R Console

> ## Z score
>
> ZC=function(d) {
+   m=mean(d)
+   s=sd(d)
+   z=(d-m)/s
+   return(z)
+ }
> d=c(72, 65, 77, 80, 73, 75, 64, 85, 70, 77)
> ZC(d)
[1] -0.2778931 -1.3585885  0.4940322  0.9571874 -0.1235080  0.1852621
[7] -1.5129736  1.7291126 -0.5866632  0.4940322
> ZC_sum=sum(ZC(d))
> ZC_sum
[1] 4.551914e-15
>

```

● Exercise 3: Find the population variance

```

> setwd("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.

```
R Console
> ## population variance
>
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_analysis.txt",header=T)
> attach(cyber_bullying)
> VAR=function(x) var(x)*(length(x)-1)/length(x)
> VAR(Onespread)
[1] 50219.9
> sqrt(VAR(Onespread))
[1] 224.098
> VAR(Twospread)
[1] 27.70627
> sqrt(VAR(Twospread))
[1] 5.263675
> |
```

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.

```
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'.



```
<-- R Console
> ## conditional statement
>
> x=c(75, 78, 80, 67, 72, 86, 62, 90, 84, 70)
> F=function(a){
+   if(a==1) { result=mean(x)
+               return(result)
+   }
+   else {
+     result=sd(x)
+     return(result)
+   }
+ }
> F(1)
[1] 76.4
> F(5)
[1] 8.871928
> |
```

- 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.”

```
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 Console
> ## iteration structure
>
> F=function(a){
+   result=0
+   for(i in 1:a){
+     result=result+i
+   }
+   return(result)
+ }
> F(100)
[1] 5050
> F(50000)
[1] 1250025000
> 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”

```
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_analysis.
txt",header=T)
    - Assign 'cyber_bullying_descriptive_analysis.txt' to cyber_bullying.
> 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

```
> 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

```
> 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 .
```



The screenshot shows an R console window with the title 'R Console'. The window contains the following R session:

```

> # variable usage($, attach, with)
>
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_analysis.txt",
+ header=T)
> cyber_bullying_1=read.table(file="cyber_bullying_descriptive_analysis_1.txt",
+ header=T)
> sd(cyber_bullying$Onespread)/mean(cyber_bullying$Onespread)
[1] 3.664652
> attach(cyber_bullying)
> sd(Onespread)/mean(Onespread)
[1] 3.664652
> attach(cyber_bullying_1)
The following objects are masked from cyber_bullying:

  Account, Channel, Onespread

> sd(Onespread)/mean(Onespread)
[1] 6.549366
> attach(cyber_bullying)
The following objects are masked from cyber_bullying_1:

  Account, Channel, Onespread

The following objects are masked from cyber_bullying (pos = 4):

  Account, Channel, Onespread, Twospread

> with(cyber_bullying_1,sd(Onespread)/mean(Onespread))
[1] 6.549366
> |

```

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.

```

> 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, 13, 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.

```

The screenshot shows the R Console window with the following content:

```

R Console
> ## R data frame write
>
> # write from vector(data.frame)
>
> V0=1:10
> V1=c(4, 7, 16, 12, 8, 11, 14, 9, 4, 8)
> 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, 13, 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)
> willard_data
   ID flaming harassment denigration impersonation outing exclusion cyberstalking
1   1        4            3          2           1         3       6        3
2   2        7            5          5           0         2       3        2
3   3       16           11          12          14        15      12       19
4   4       12           11          17          12        13      10       15
5   5        8            6          9           0         8       3        7
6   6       11           6          6           0         2       4        8
7   7       14           13          15          4         6       5        14
8   8        9            4          6           1         2       6        4
9   9        4            3          3           0         1       3        2
10 10       8            7          9           1         4       5        8
> |

```

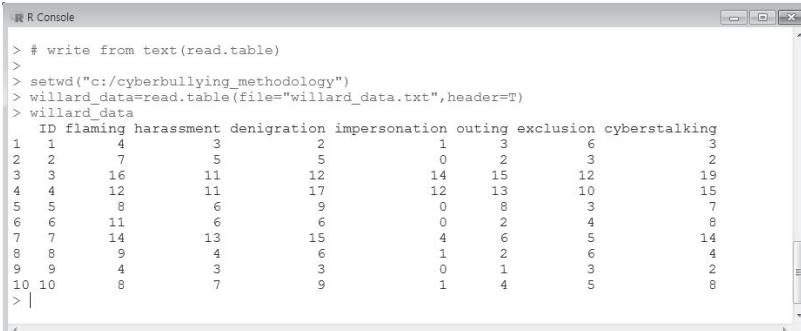
(2) Creating data frames from text files

- 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.

```



The screenshot shows the R Console window with the following code and output:

```

> # write from text(read.table)
>
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data
   ID flaming harassment denigration impersonation outing exclusion cyberstalking
1  1          4             3            2           1         3        6          3
2  2          7             5            5           0         2        3          2
3  3         16            11            12          14        15        12         19
4  4          12            11            17          12        13        10         15
5  5          8              6            9           0         8        3          7
6  6          11            6             6           0         2        4          8
7  7          14            13            15          4         6        5         14
8  8          9              4             6           1         2        6          4
9  9          4              3             3           0         1        3          2
10 10         8              7             9           1         4        5          8
> |

```

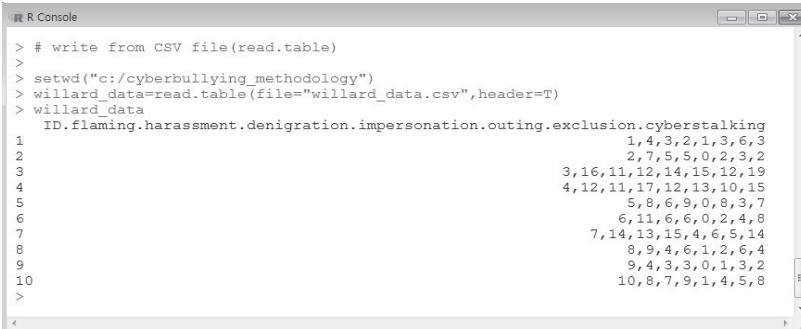
(3) Creating data frames from CSV files

- Use the `read.table()` function.

```

> setwd("c:/cyberbullying_methodology") : Set the working directory.
> willard_data=read.table(file="willard_data.csv",header=T)
  - Assign "willard _data.csv" to the willard_data object via the data
    frame.
> willard_data : Display the value of the willard_data object on the screen.

```



The screenshot shows the R Console window with the following code and output:

```

> # write from CSV file(read.table)
>
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.csv",header=T)
> willard_data
   ID.flaming.harassment.denigration.impersonation.outing.exclusion.cyberstalking
1          1,4,3,2,1,3,6,3
2          2,7,5,5,0,2,3,2
3          3,16,11,12,14,15,12,19
4          4,12,11,17,12,13,10,15
5          5,8,6,9,0,8,3,7
6          6,11,6,6,0,2,4,8
7          7,14,13,15,4,6,5,14
8          8,9,4,6,1,2,6,4
9          9,4,3,3,0,1,3,2
10         10,8,7,9,1,4,5,8
>

```

(4) Creating data frames from SPSS files

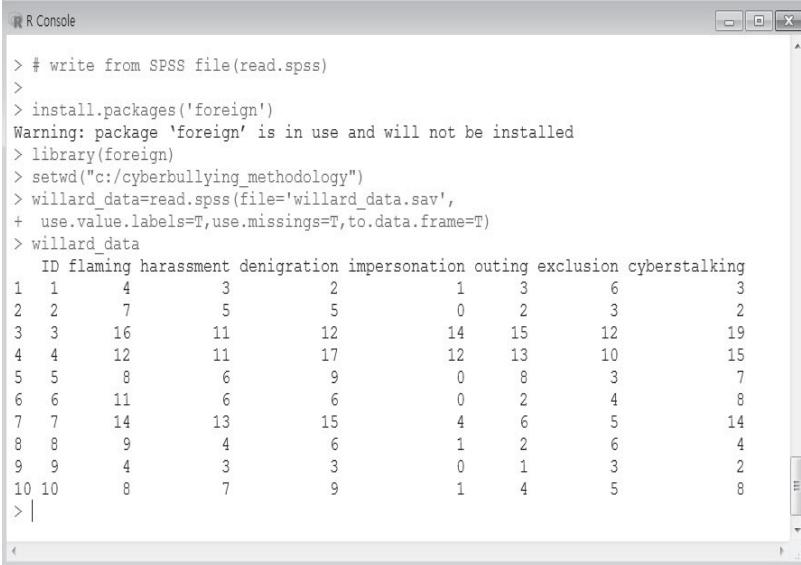
- Use the `read.spss()` function.

```

> install.packages('foreign')
  - Install a package for reading external data created by statistics
    software other than R; e.g., SPSS or SAS.

```

```
> library(foreign) : Load the foreign package.  
> setwd("setwd("c:/cyberbullying_methodology")  
      - Set the working directory.  
> willard_data=read.spss(file='willard_data.sav',use.value.labels=T,use.  
missings=T,to.data.frame=T)  
      - Assign "willard_data.sav" to the willard_data object via the data  
frame.  
      - file=' ' : Define the external files from which the data will be read in.  
      - use.value.labels = T : Define the labels defined in the external data's  
variable values as the variable labels of the R data frame.  
      - use.missings = T : Define whether there are missing values used in  
the external data variables.  
      - to.data.frame = T : Define whether it is created as a data frame.  
> willard_data : Display the value of the willard_data object on the screen.
```



The screenshot shows the R Console window with the following content:

```
R Console
```

```
> # write from SPSS file(read.spss)
>
> install.packages('foreign')
Warning: package 'foreign' is in use and will not be installed
> library(foreign)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.spss(file='willard_data.sav',
+ use.value.labels=T,use.missings=T,to.data.frame=T)
> willard_data
   ID flaming harassment denigration impersonation outing exclusion cyberstalking
1  1        4            3          2          1          3          6          3
2  2        7            5          5          0          2          3          2
3  3       16           11          12         14         15          12         19
4  4       12           11          17         12         13          10         15
5  5        8            6          9          0          8          3          7
6  6       11           6          6          0          2          4          8
7  7       14           13          15          4          6          5         14
8  8        9            4          6          1          2          6          4
9  9        4            3          3          0          1          3          2
10 10       8            7          9          1          4          5          8
> |
```

The data frame contains 10 rows and 8 columns, with the first row being the header. The columns are labeled: ID, flaming, harassment, denigration, impersonation, outing, exclusion, and cyberstalking.

ID	flaming	harassment	denigration	impersonation	outing	exclusion	cyberstalking
1	1	4	3	2	1	3	6
2	2	7	5	5	0	2	3
3	3	16	11	12	14	15	12
4	4	12	11	17	12	13	10
5	5	8	6	9	0	8	3
6	6	11	6	6	0	2	4
7	7	14	13	15	4	6	5
8	8	9	4	6	1	2	6
9	9	4	3	3	0	1	3
10	10	8	7	9	1	4	5

(5) Displaying a data frame from a text file

- Use the write.matrix() function.

```
> setwd("c:/cyberbullying_methodology") : Set the working directory.
> willard_data=read.table(file="willard_data.txt",header=T)
      - Assign "willard_data.txt" to the willard_data object via the data frame.
```

```
> willard_data : Display the willard_data object's values on the screen.
> library(MASS) : Load the package for using the write.matrix( ) function.
> write.matrix(willard_data, "willard_data_w.txt")
  - Write the willard_data object to the willard_data_w.txt" file.
> willard_data_w= read.table('willard_data_w.txt',header=T) : Read the
  "willard_data_w.txt" file and save it in the willard_data_w object.
> willard_data_w
  - Display the value of the willard_data_w object on the screen.
```

	ID	flaming	harassment	denigration	impersonation	outing	exclusion	cyberstalking
1	1	4	3	2	1	3	6	3
2	2	7	5	5	0	2	3	2
3	3	16	11	12	14	15	12	19
4	4	12	11	17	12	13	10	15
5	5	8	6	9	0	8	3	7
6	6	11	6	6	0	2	4	8
7	7	14	13	15	4	6	5	14
8	8	9	4	6	1	2	6	4
9	9	4	3	3	0	1	3	2
10	10	8	7	9	1	4	5	8

	ID	flaming	harassment	denigration	impersonation	outing	exclusion	cyberstalking
1	1	4	3	2	1	3	6	3
2	2	7	5	5	0	2	3	2
3	3	16	11	12	14	15	12	19
4	4	12	11	17	12	13	10	15
5	5	8	6	9	0	8	3	7
6	6	11	6	6	0	2	4	8
7	7	14	13	15	4	6	5	14
8	8	9	4	6	1	2	6	4
9	9	4	3	3	0	1	3	2
10	10	8	7	9	1	4	5	8

(6) Combining files [Combining variables (Columns)]

- Use the write.matrix() and cbind() functions.

```
> library(MASS) : Load the package for using the write.matrix( ) function.
> setwd("c:/cyberbullying_methodology") : Set the working directory.
> willard_data=read.table(file="willard_data.txt",header=T)
  - Assign "willard_data.txt" to the willard_data object via the data frame.
> willard_data_1=read.table(file="willard_data_1.txt",header=T) : Assign
  "willard_data_1.txt" to the willard_data_1 object via the data frame
> willard_data
  - Display the values of the willard_data objects on the screen.
```

```
> willard_data_1
> willard_data_ac=cbind(willard_data,willard_data_1$denigration_1)
  - Combine the variables (denigration_1) set in willard_data and
    willard_data_1 and save them in willard_data_ac.
> willard_data_ac
  - Display the value of the willard_data_ac object on the screen.
> willard_data_ac=cbind(willard_data,willard_data_1)
  - Combine all variables in willard_data and willard_data_1 and save
    them in willard_data_ac.
> write.matrix(willard_data_ac, "willard_data_ac.txt")
  - Write the willard_data_ac object to the "willard_data_ac.txt" file.
```

The screenshot shows the R Console window with the following content:

```
> ## file combine [variable(column)combine] - cbind()
>
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data_1=read.table(file="willard_data_1.txt",header=T)
> willard_data
  ID flaming harassment denigration impersonation outing exclusion cyberstalking
1 1 4 3 2 1 3 6 3
2 2 7 5 5 0 2 3 2
3 3 16 11 12 14 15 12 19
4 4 12 11 17 12 13 10 15
5 5 8 6 9 0 8 3 7
6 6 11 6 6 0 2 4 8
7 7 14 13 15 4 6 5 14
8 8 9 4 6 1 2 6 4
9 9 4 3 3 0 1 3 2
10 10 8 7 9 1 4 5 8
> willard_data_1
  ID flaming_1 harassment_1 denigration_1 impersonation_1 outing_1 exclusion_1 cyberstalking_1
1 1 4 3 2 1 3 6 3
2 2 7 5 5 0 2 3 2
3 3 16 11 12 14 15 12 19
4 4 12 11 17 12 13 10 15
5 5 8 6 9 0 8 3 7
6 6 11 6 6 0 2 4 8
7 7 14 13 15 4 6 5 14
8 8 9 4 6 1 2 6 4
9 9 4 3 3 0 1 3 2
10 10 8 7 9 1 4 5 8
> willard_data.ac=cbind(willard_data,willard_data_1$denigration_1)
> willard_data.ac
  ID flaming harassment denigration impersonation outing exclusion cyberstalking willard_data_1$denigration_1
1 1 4 3 2 1 3 6 3 2
2 2 7 5 5 0 2 3 2 5
3 3 16 11 12 14 15 12 19 12
4 4 12 11 17 12 13 10 15 17
5 5 8 6 9 0 8 3 7 9
6 6 11 6 6 0 2 4 8 6
7 7 14 13 15 4 6 5 14 15
8 8 9 4 6 1 2 6 4 6
9 9 4 3 3 0 1 3 2 3
10 10 8 7 9 1 4 5 8 9
> willard_data.ac=cbind(willard_data,willard_data_1)
> write.matrix(willard_data.ac, "willard_data_ac.txt")
> |
```

(7) Combining files [Combining records (Rows)]

- Use the write.matrix() and rbind() functions.

```
> library(MASS)
> setwd("c:/cyberbullying_methodology")
```

```
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data_2=read.table(file="willard_data_2.txt",header=T)
> willard_data_ar=rbind(willard_data,willard_data_2)
  - The variable names within the files must be the same.
  - Add the records of willard_data_2 to the willard_data data file and
    save it in willard_data_ar.
> willard_data_ar
> write.matrix(willard_data_ar, "willard_data_ar.txt")
```

The screenshot shows the R Console window with the following content:

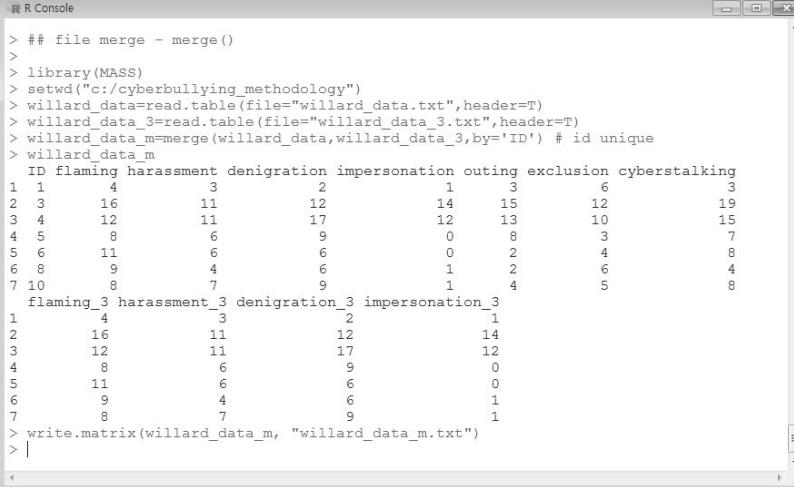
```
> ## file combine [record(row)combine] - rbind()
>
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data_2=read.table(file="willard_data_2.txt",header=T)
> willard_data_ar=rbind(willard_data,willard_data_2)
> willard_data_ar
   ID flaming harassment denigration impersonation outing exclusion cyberstalking
 1  1      4            3          2        1     3       6           3
 2  2      7            5          5        0     2       3           2
 3  3     16           11         12       14     15      12        19
 4  4     12           11         17       12     13      10        15
 5  5      8            6          9        0     8       3           7
 6  6     11           6          6        0     2       4           8
 7  7     14           13         15       4     6       5        14
 8  8      9            4          6        1     2       6           4
 9  9      4            3          3        0     1       3           2
10 10     8            7          9        1     4       5           8
11 11     4            3          2        1     3       6           3
12 12     7            5          5        0     2       3           2
13 13     16           11         12       14     15      12        19
14 14     12           11         17       12     13      10        15
15 15     5            8          6          9        0     8       3           7
16 16     6           11           6          6        0     2       4           8
17 17     14           13         15       4     6       5        14
18 18     8            9          4          6        1     2       6           4
19 19     9            4          3          3        0     1       3           2
20 20     10           8           7          9        1     4       5           8
> write.matrix(willard_data_ar, "willard_data_ar.txt")
> |
```

(8) Merging files [Combining with the same ID]

- Use the write.matrix() and merge() functions.

```
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data_3=read.table(file="willard_data_3.txt",header=T)
> willard_data_m=merge(willard_data,willard_data_3,by='ID') # id unique
  - Merge the willard_data data and the willard_data_3 data that have the
    same IDs, and save them in willard_data_m.
> willard_data_m
> write.matrix(willard_data_m, "willard_data_m.txt")
```

- It can be seen that only the same IDs (1, 3, 4, 5, 6, 8, 10) were merged.



The screenshot shows an R console window with the title "R Console". The code in the console is as follows:

```
> ## file merge - merge()
>
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data_3=read.table(file="willard_data_3.txt",header=T)
> willard_data_m=merge(willard_data,willard_data_3,by='ID') # id unique
> willard_data_m
```

The output displays two data frames. The first data frame has columns ID, flaming, harassment, denigration, impersonation, outing, exclusion, and cyberstalking. The second data frame has columns flaming_3, harassment_3, denigration_3, impersonation_3. Both data frames have rows corresponding to the IDs 1, 3, 4, 5, 6, 8, and 10. The data values are as follows:

ID	flaming	harassment	denigration	impersonation	outing	exclusion	cyberstalking	
1	1	4	3	2	1	3	6	3
2	3	16	11	12	14	15	12	19
3	4	12	11	17	12	13	10	15
4	5	8	6	9	0	8	3	7
5	6	11	6	6	0	2	4	8
6	8	9	4	6	1	2	6	4
7	10	8	7	9	1	4	5	8

	flaming_3	harassment_3	denigration_3	impersonation_3
1	4	3	2	1
2	16	11	12	14
3	12	11	17	12
4	8	6	9	0
5	11	6	6	0
6	9	4	6	1
7	8	7	9	1

(9) Selecting Variables and Observed Values

- Select variables

```
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data
> attach(willard_data)
> willard_data_v=data.frame(denigration,outing,cyberstalking)
  - Select only the variables set in willard_data (denigration, outing,
    cyberstalking) and save them in willard_data_v.
> willard_data_v
> write.matrix(willard_data_v, "willard_data_vw.txt")
```

```

> # variable selection
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data
ID flaming harassment denigration impersonation outing exclusion cyberstalking
1 1 4 3 2 1 3 6 3
2 2 7 5 5 0 2 3 2
3 3 16 11 12 14 15 12 19
4 4 12 11 17 12 13 10 15
5 5 8 6 9 0 8 3 7
6 6 11 6 6 0 2 4 8
7 7 14 13 15 4 6 5 14
8 8 9 4 6 1 2 6 4
9 9 4 3 3 0 1 3 2
10 10 8 7 9 1 4 5 8
> attach(willard_data)
> willard_data_v=data.frame(denigration,outing,cyberstalking)
> willard_data_v
denigration outing cyberstalking
1 2 3 3
2 5 2 2
3 12 15 19
4 17 13 15
5 9 6 7
6 6 2 8
7 15 6 14
8 6 2 4
9 3 1 2
10 9 4 8
> write.matrix(willard_data_v, "willard_data_vw.txt")

```

● Select observed values

```

> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data
> attach(willard_data)
> willard_data_c=willard_data[willard_data$flaming!=4,]
- Select only the rows where the value of willard_data's flaming
  variable is not 4 and save them in willard_data_c.
> willard_data_c
> write.matrix(willard_data_c, "willard_data_cw.txt")

```

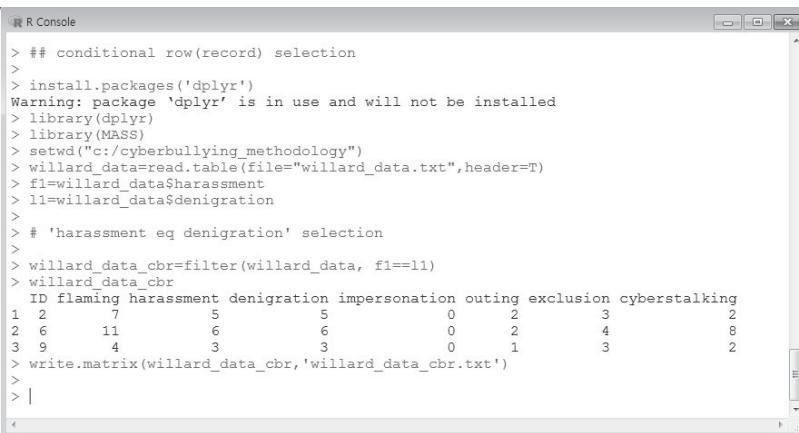
```

> # value selection
>
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> willard_data
ID flaming harassment denigration impersonation outing exclusion cyberstalking
1 1 4 3 2 1 3 6 3
2 2 7 5 5 0 2 3 2
3 3 16 11 12 14 15 12 19
4 4 12 11 17 12 13 10 15
5 5 8 6 9 0 8 3 7
6 6 11 6 6 0 2 4 8
7 7 14 13 15 4 6 5 14
8 8 9 4 6 1 2 6 4
9 9 4 3 3 0 1 3 2
10 10 8 7 9 1 4 5 8
> attach(willard_data)
The following objects are masked from willard_data (pos = 3):
  cyberstalking, denigration, exclusion, flaming, harassment, ID,
  impersonation, outing
> willard_data_c=willard_data[willard_data$cflaming!=4,]
> willard_data_c
ID flaming harassment denigration impersonation outing exclusion cyberstalking
2 2 7 5 5 0 2 3 2
3 3 16 11 12 14 15 12 19
4 4 12 11 17 12 13 10 15
5 5 8 6 9 0 8 3 7
6 6 11 6 6 0 2 4 8
7 7 14 13 15 4 6 5 14
8 8 9 4 6 1 2 6 4
9 9 4 3 3 0 1 3 2
10 10 8 7 9 1 4 5 8
> write.matrix(willard_data_c, "willard_data_cw.txt")

```

- Extract rows (Records) according to conditions

```
> install.packages('dplyr')
> library(dplyr)
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> f1=willard_data$harassment
> l1=willard_data$denigration
> willard_data_cbr=filter(willard_data, f1==l1)
  - Extract only the "harassment equal denigration" rows and save them
    to willard_data_cbr
> willard_data_cbr
> write.matrix(willard_data_cbr,'willard_data_cbr.txt')
```



The screenshot shows the R Console window with the following content:

```
R Console
> ## conditional row(record) selection
>
> install.packages('dplyr')
Warning: package 'dplyr' is in use and will not be installed
> library(dplyr)
> library(MASS)
> setwd("c:/cyberbullying_methodology")
> willard_data=read.table(file="willard_data.txt",header=T)
> f1=willard_data$harassment
> l1=willard_data$denigration
>
> # 'harassment eq denigration' selection
>
> willard_data_cbr=filter(willard_data, f1==l1)
> willard_data_cbr
  ID flaming harassment denigration impersonation outing exclusion cyberstalking
1  2        7           5          5         0       2      3        2
2  6       11          6           6         0       2      4        8
3  9        4           3          3         0       1      3        2
> write.matrix(willard_data_cbr,'willard_data_cbr.txt')
>
> |
```

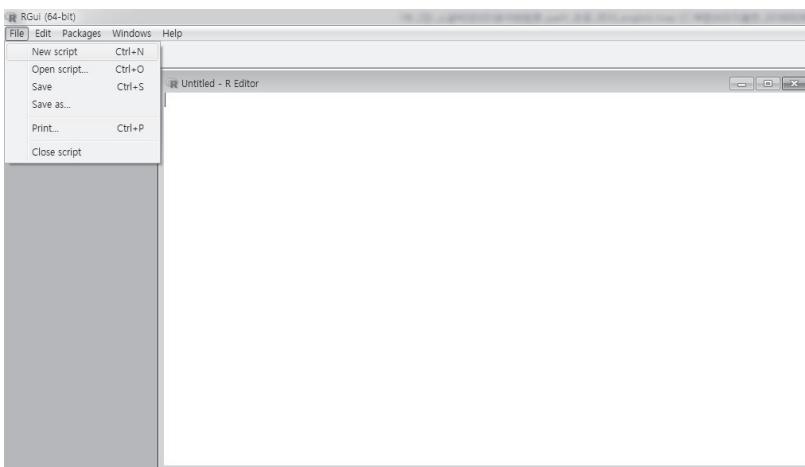
The output table has the following structure:

ID	flaming	harassment	denigration	impersonation	outing	exclusion	cyberstalking	
1	2	7	5	5	0	2	3	2
2	6	11	6	6	0	2	4	8
3	9	4	3	3	0	1	3	2

7) Using R's Main GUI (Graphic User Interface) Menus

(1) Create New Script: [File → New script]

- After scripts are created in the R-Editor, they can be opened and run in the R-Console screen.



(2) Save New Script: [File → Save as...]

```

## maching learning wordcloud 2018. 5. 10.

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

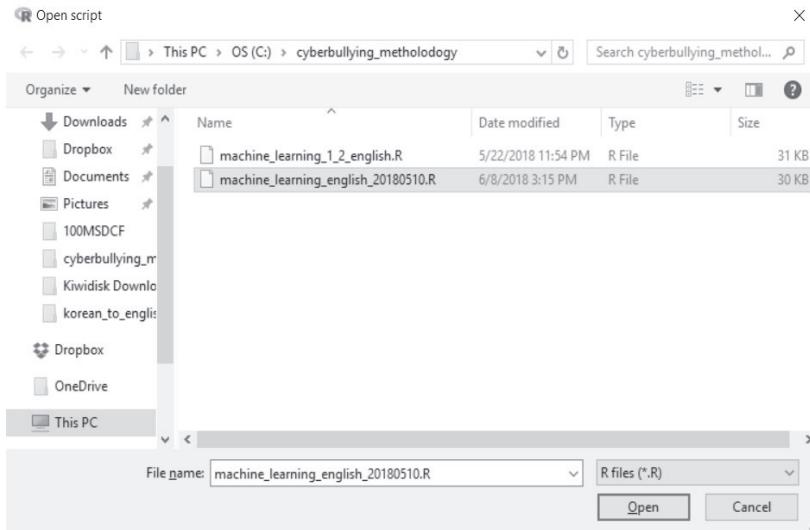
key=c('Domestic-violence','Child-abuse','Parental-divorce','Economic-problems',
      'Friend-Violence','Break-ups','School-control','Academic-stress','School_recor
      'School-violence-experience','Transfer','Individualism','Materialism','Bullyin
      'Class-society','Hell-Korea','Female_dislike','Interested_soldier','Traffic-ac
      'Games','Internet-addiction','Celebrities','Movie','Adults','Gags','Chat-apps'
      'Personal-broadcasting')

freq=c(2269,1338,3515,7269,5844,1101,32816,1503,32084,5849,8949,2348,858,539,61
      6452,784,1852,1764,2496,29473,24413,488,799,2253,1497,1153)
library(RColorBrewer)
palette=brewer.pal(9,"Set1")
wordcloud(key,freq,scale=c(4,1),rot.per=.12,min.freq=100,random.order=F,
          random.color=T,colors=palette)
savePlot ("cyber_bullying_strain_wordcloud",type="png")

```

※ All scripts used in this chapter are saved in ‘machine_learning.R’.

(3) Open New Script: [File → Open script...]



(4) Execute Script

- Select the scripts to be executed in the script editor and execute them with “CTRL +R”.

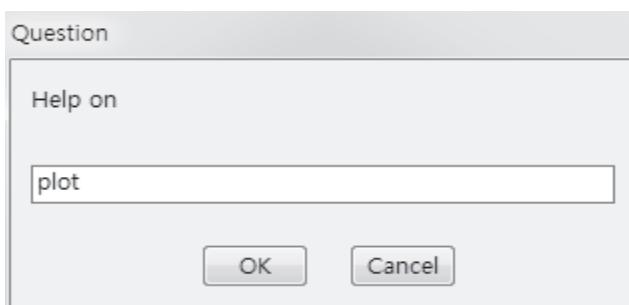
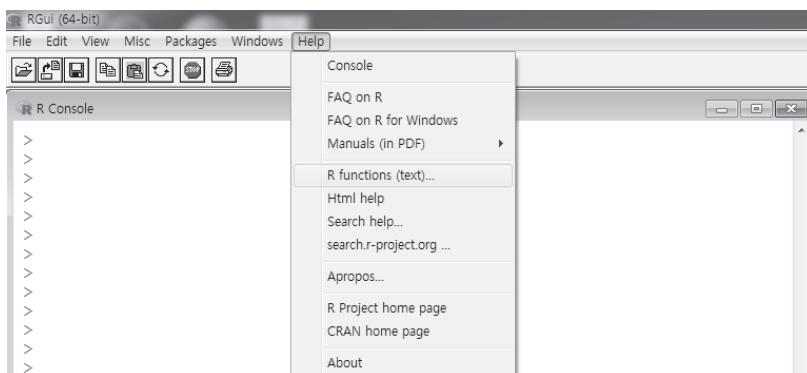
```
R C:\cyberbullying_methodology\#machine_learning_english_20180510.R - R Editor
## machine learning wordcloud 2018. 5. 10.

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

key=c('Domestic-violence','Child-abuse','Parental-divorce','Economic-problems',
      'Friend-Violence','Break-ups','School-control','Academic-stress','School_records',
      'School-violence-experience','Transfer','Individualism','Materialism','Bullying-culture',
      'Class-society','Hell-Korea','Female_dislike','Interested_soldier','Traffic-accidents',
      'Games','Internet-addiction','Celebrities','Movie','Adults','Gags','Chat-apps','Youtube',
      'Personal-broadcasting')

freq=c(2269,1338,3515,7269,5844,1101,32816,1503,32084,5849,8949,2348,859,539,617,1085,
      6452,784,1852,1764,2496,29473,24413,488,799,2253,1497,1153)
library(RColorBrewer)
palete=brewer.pal(9,"Set1")
wordcloud(key,freq,scale=c(4,1),rot.per=.12,min.freq=100,random.order=F,
          random.color=T,colors=palete)
savePlot("cyber_bullying strain wordcloud",type="png")
```

(5) Use R's Help Function: [Help → R functions (text)...]



- If “help” is entered as the input for the `plot()` function, detailed help information on the `plot` function and its arguments can be found.

```
plot [graphics]

Generic X-Y Plotting

Description
Generic function for plotting of R objects. For more details about the graphical parameter arguments, see par.
For simple scatter plots, plot.default will be used. However, there are plot methods for many R objects, including functions, data.frames, density objects, etc. Use methods(plot) and the documentation for these.

Usage
plot(x, y, ...)

Arguments
x
  the coordinates of points in the plot. Alternatively, a single plotting structure, function or any R object with a plot method can be provided.
y
  the y coordinates of points in the plot, optional if x is an appropriate structure.
...
  Arguments to be passed to methods, such as graphical parameters (see par). Many methods will accept the following arguments:
```

SCIENTIFIC RESEARCH DESIGN

Science is a systematic enterprise that builds and organizes knowledge in the form of testable explanations and predictions about the universe. Natural science is a branch of science concerned with the description, prediction, and understanding of natural phenomena, based on empirical evidence from observation and experimentation. Social science is a major category of academic disciplines, concerned with society and the relationships among individuals within a society (<https://en.wikipedia.org/wiki/> (accessed on May 10, 2018)).

To gain scientific knowledge, problems about phenomena should be conceptualized into hypotheses, which must then be tested. Thus, to solve problems using scientific thought, it is necessary to employ logical reasoning and infer principles based on empirical testing. Scientific inference includes deductive and inductive methods(Table 1).

To conduct scientific research, the researcher must pose questions regarding social phenomena, establish the research goals and topics, and conduct a literature review. Based on these, the researcher must set up a research model and formulate the hypotheses to be tested. After developing measurement tools during the survey-design stage and conducting sampling, the researcher must collect and analyze the data before arriving at a conclusion.

Table 1 Deductive and Inductive Inference

Scientific Inference Method	Definition & Characteristics
Deductive	<ul style="list-style-type: none">- A method of inferring specific facts or statements based on more general facts or previous theories.- Follows the process of ‘theory → hypothesis → facts’.- Related to the concept of confirmatory factor analysis, where theoretical results are inferred. e.g., ‘All men shall die’ → ‘Socrates is a man’ → ‘Therefore, Socrates shall die.’

Inductive	<ul style="list-style-type: none"> - A method of inferring general facts based on facts or specific cases as observed by the researcher. - Follows the process of ‘facts → exploration → theory’. - Related to the concept of exploratory factor analysis, conducted to establishing a research direction in the absence of previous hypotheses or theories about latent factors. - Machine learning is an inductive inference method, where data is used to train a specific model, which is then generalized via abstraction. <p>e.g., ‘Socrates has died, Confucius has died, as have ...’ → ‘All of these persons are men’ → ‘Therefore, all men shall die.’</p>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Research Concepts

A concept is an abstract idea that refers to some phenomenon. In scientific research models, concepts are used as constructs, which may be defined conceptually or operationally.

Table 2 Research Concepts

Category	Definition & Characteristics
Conceptual definition	<ul style="list-style-type: none"> • A previously agreed-upon concept, abstractly expressed in the form of language, which refers to some idea that is to be examined. e.g., ‘Self-Esteem’
Operational definition	<ul style="list-style-type: none"> • A concrete statement that ties a conceptual definition to actually observable (measurable) phenomena. e.g., Self-Esteem: Rosenberg Self Esteem Scales - I feel that I am a person of worth, at least on an equal plane with others. - I feel that I have a number of good qualities. - All in all, I am inclined to feel that I am a failure. - I am able to do things as well as most other people. - I feel I do not have much to be proud of. - I take a positive attitude toward myself. - On the whole, I am satisfied with myself. - I wish I could have more respect for myself. - I feel completely useless at times. - At times I think I am no good at all.

Variable Measurement

To conduct scientific research, it is necessary to collect appropriate data and then discern whether those data are suitable for statistical analysis. Measurement refers to the act of methodically assigning numerical values to the characteristics of empirically observed objects or phenomena, in accordance with certain rules. The measurement rule, or scale, is a method of assigning numerical values to some object. Variables represent the properties or characteristics of some measured object or phenomenon, and refer to sub-concepts that may be used for assigning operational definitions to empirical concepts.

Scale

Scales refer to measurement units for concretizing the properties of variables. As described in Table 3, scales largely vary in precision from nominal, ordinal, and interval scales to ratio scales. Alternatively, as in Fig. 1, data may be categorized as either qualitative or quantitative, depending on their properties.

Table 3 Scales Categorization, by Measurement Precision

Category	Definition & Characteristics
Nominal scale	<ul style="list-style-type: none">- Variable values are categorized or assigned names; their properties are not quantitative but determined by type or quality. e.g., place of residence, marital status, religion, illness, etc.
Ordinal scale	<ul style="list-style-type: none">- Variables have graduated values, which are determined by their properties. e.g., level of education, social status, academic rank, order of service preference, etc.
Interval scale	<ul style="list-style-type: none">- Orders may be assigned depending on the quantity of some variable properties.- By assigning the same units to intervals of the same size, the scale is equidistant. The scale has no absolute-zero point, and ratios between values are not meaningful. Thus, the scale admits only additive operations. e.g., temperature, IQ score, stock price indices, etc.
Ratio scale	<ul style="list-style-type: none">- This scale augments the interval scale by making ratios meaningful. It has an absolute-zero point and arbitrary units, and admits both additive and multiplicative operations. e.g., body weight, height, age, income, sales revenue, etc.

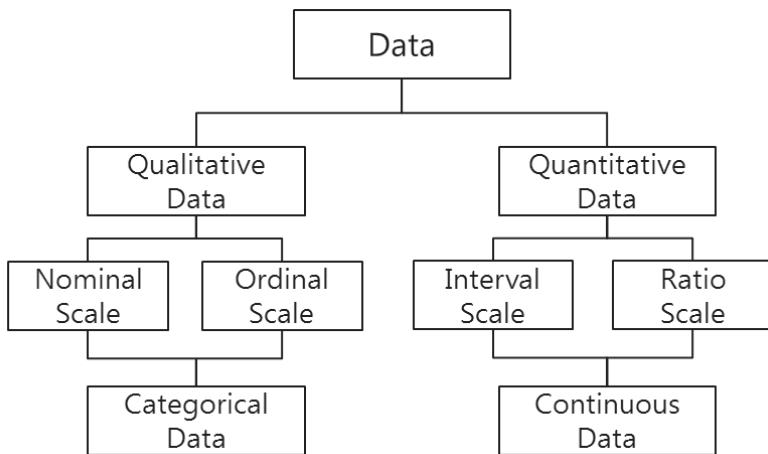


Fig. 1. Data Categorization, by Scale Properties

Variables

Variables refer to all numbers whose values may vary depending on different conditions, taking on at least two values. On the other hand, constants refer to fixed numbers whose values do not change. Depending on the causal links between variables, they may be categorized as independent, dependent, mediating, or moderation.

Independent variables refer to variables that influence other variables; they are alternately known as predictor, explanatory, cause, or covariate variables. The dependent variable refers to the variable that is influenced by the independent variables; it is also known as the response or effect variable. The mediating variable refers to a variable that works between the independent and dependent variables; i.e., it is influenced by the independent variables while also influencing the dependent variable. It refers to a variable that must be controlled for in the research. Therefore, the mediating effect occurs when a third mediating variable intervenes between the dependent and independent variables (Baron & Kenny, 1986).

A moderation variable refers to a third variable that changes the nature of the relationships between variables (e.g., between the dependent and independent variables). It may change the directionality or strength of the relationships between variables.

For instance, if self-control influences the pathway from strain to delinquency, strain is the independent variable, delinquency is the

dependent variable, and self-control is the mediating variable. If there are gender differences in the pathway from strain to delinquency, strain is the independent variable, delinquency is the dependent variable, and gender is the moderation variable.

Unit of Analysis

The unit of analysis is the basic unit that determines the sample size—it may either be individuals, groups, or certain organizations. It can also determine the level of analysis. The researcher may, in practice, choose to go into further detail beyond the direct research subjects into the lower levels, or may instead choose to aggregate results to higher levels. Thus, it is the basic unit of data analysis.

Some of the fallacies that result from incorrect inferences regarding units of analysis are the ecological fallacy, individualistic fallacy, and the reductionism fallacy.

The ecological fallacy may occur when estimating the characteristics of individuals belonging to some group, based on that group's characteristics (e.g., analyzing the characteristics of Catholics as a group, and basing the analysis of individual Catholics on those characteristics). Conversely, the individualistic fallacy may occur when estimating the characteristics of groups based on the characteristics of individuals who constitute that group (e.g., finding that individuals from some society have a strong sense of public order, and using this as the basis to conclude that that society has a strong sense of public order).

The reductionism fallacy includes the individualistic fallacy, and refers to the tendency to employ variables in unduly limited or reductionist manners (e.g., a psychologist who uses only psychological variables in examining social phenomena, despite the need for a multifaceted approach that also includes economic or political variables). Thus, the individualistic fallacy pertains to the unit of analysis, while the reductionism fallacy pertains to the selection of variables.

Sampling and Hypothesis Testing

Sampling

In the scientific survey process, once the measurement instruments are established, it is necessary to decide the scope of the data collection: whether to survey the subjects in full (population) or in part (sample).

The population refers to the entire group of the study subjects, where

the goal of scientific research is to describe or infer the characteristics of the population. Studying the entire population is often infeasible because of prohibitive costs (economic reasons) or insufficient time (temporal reasons). Therefore, when researchers aim to gather information or knowledge about some population, partial samples are drawn from the population and then analyzed to infer characteristics about the population.

As illustrated in Fig. 2, parameters—e.g., the population mean (μ), standard deviation (σ), and correlation coefficient (ρ)—refer to the characteristic values of the population. Statistics—e.g., the sample mean (\bar{x}), sample standard deviation (s), and sample correlation coefficient (r)—represent the characteristic values of the sample.

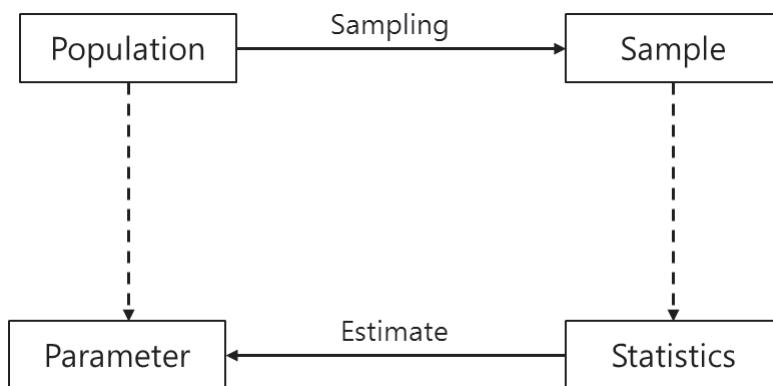


Fig. 2. Relationship between Population and Sample Studies

To extract a sample from the population, the researcher must determine the sample size that will preserve the representativeness of the sample. The sample size is determined based on the nature of the population, research objectives, and economic and temporal costs. Generally, in the case of opinion polls, it is possible to determine the sample size based on the level of confidence and the sampling error (the expected error arising from differences in the population during the sampling process).

Sampling methods largely consist of probability and non-probability sampling. Probability sampling refers to a sampling method where the selection probabilities of all population elements are known. Simple random sampling, stratified sampling, and cluster sampling fall under into this category.

In simple random sampling, every element of the population has the

same probability of selection. In systematic sampling, each element of the population is enumerated, and each k^{th} element is selected into the sample.

In stratified sampling, the population is partitioned, according to some rule, into several homogenous strata, after which simple random sampling is applied to each stratum. In cluster sampling, the population is grouped, according to some rule, into several clusters. After some of the clusters are selected at random, the elements of these clusters are selected as the sample. Stratified random-cluster sampling combines the stratified, cluster, and simple probability sampling methods.

Non-probability sampling is used when the selection probability of each population element is unknown. Convenience sampling, judgment sampling, snowball sampling, and quota sampling fall into this category. Convenience sampling is where the sample is selected at the researcher's convenience; it is also known as accidental sampling. Judgment sampling is a method of deliberately selecting a certain group that is thought to reflect the views of the population; it is also known as purposive sampling. In quota sampling, the population is subdivided according to some predetermined rule, after which the sample is selected according to the ratio of each group's share among the population. In snowball sampling, some elements of the population are initially selected into the sample, after which other elements are iteratively selected, upon the recommendations of the initially selected elements.

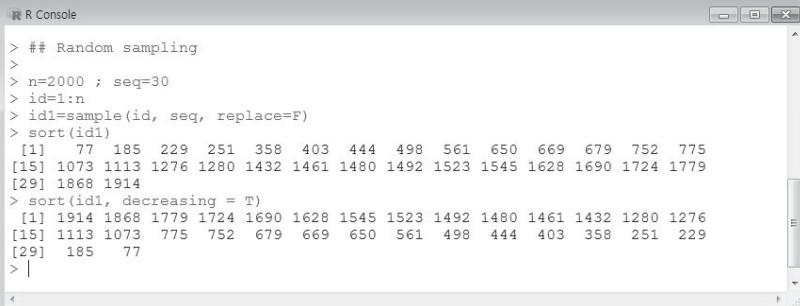
■ Example: Random Sampling Method

- To randomly hand out prizes to 30 persons from among 2,000 survey respondents, it is necessary to randomly select 30 persons from the 2,000.
- Use the `sample()` function in R. From among the elements of a given vector of length n , randomly select a sub-vector of length seq .

```
> n=2000 ; seq=30
- Assign 2000 to n and 30 to seq.
> id=1:n
- Assign values from 1 to 2000 to the vector 'id'.
> id1=sample(id, seq, replace=F)
- Randomly select 30 persons from vector 'id' and assign them to the vector 'id1'.
- replace = F (sampling without replacement), replace = T (sampling with replacement: some elements may be selected repeatedly).
> sort(id1): Sort the randomly selected vector 'id1' in increasing order.
```

```
> sort(id1, decreasing = T)
```

- Sort the randomly selected vector ‘id1’ in decreasing order.



```
R Console

> ## Random sampling
>
> n=2000 ; seq=30
> id=1:n
> id1=sample(id, seq, replace=F)
> sort(id1)
[1]  77 185 229 251 358 403 444 498 561 650 669 679 752 775
[15] 1073 1113 1276 1280 1432 1461 1480 1492 1523 1545 1628 1690 1724 1779
[29] 1868 1914
> sort(id1, decreasing = T)
[1] 1914 1868 1779 1724 1690 1628 1545 1523 1492 1480 1461 1432 1280 1276
[15] 1113 1073 775 752 679 669 650 561 498 444 403 358 251 229
[29] 185   77
> |
```

Hypothesis Testing

To conduct scientific research, the researcher must first review various studies and reports regarding the topic of interest in order to establish the likely causal links between concepts. Then, based on previous theories and the researcher’s empirical observations, a research model must be specified, after which hypotheses should be introduced and tested, based on the model.

A hypothesis is a tentative statement regarding the research topic. The hypothesis is tested to estimate the population parameters based on the statistics calculated from the sample. Hypothesis testing is an inference process where the researcher determines the rejection region of the sampling error between the statistics and the parameters. Therefore, because parameter estimates are not identical, a confidence interval is used to determine whether a hypothesis is accepted. When estimating a population parameter based on a sample statistic, the confidence interval indicates the region that is expected to include the parameter.

A hypothesis can be largely categorized as a null hypothesis (H_0) and an alternative (research) hypothesis (H_1). The null hypothesis usually takes the form of ‘the parameter equals some value’ or ‘the two parameters are the same as (not different from) each other,’ while the alternative hypothesis takes the form of ‘the parameter does not equal some value’ or ‘the two parameters are unequal (one is smaller or larger than the other).’ Thus, the null hypothesis indicates no departure from previous or general facts, while the alternative hypothesis indicates that the researcher has discovered some new fact that departs from previous or general knowledge. Therefore, in hypothesis testing, the goal is to test whether there is a

significant difference in the estimates from the sample.

Because it is theoretically impossible to perfectly test hypotheses, two types of error arise. A type-I error (α) refers to the rejection of H_0 when it is true (i.e., stating that an effect exists when in fact it does not), while a type-II error (β) refers to the acceptance of H_0 when it is false (i.e., stating that an effect does not exist when in fact it does). Hypothesis testing is conducted by determining whether the null or alternative hypothesis should be rejected, based on a comparison of the sample statistic—the p -value—and the level of the type-I error—the level of significance. The p -value is a statistic that is computed from the sample; it indicates the probability of observing the given sample under the null hypothesis. The significance level, expressed as ' α ', is chosen by the researcher to determine the threshold level that the p -value must reach for the null hypothesis to be rejected and the alternative hypothesis to be accepted. Common choices for the level of significance include .001, .01, .05, and .1.

In hypothesis testing, the null hypothesis is rejected when ' $p < \alpha$ '. Thus, if hypothesis testing is conducted under ' $p < .05$ ', the researcher will allow for at most a 5% probability of a type-I error; i.e., there is at least a 95% chance that the alternative hypothesis is true. Thus, statistical inference involves the inference of population characteristics based on the analysis of sample characteristics, with the outcomes determined by means of hypothesis testing (Fig. 3).

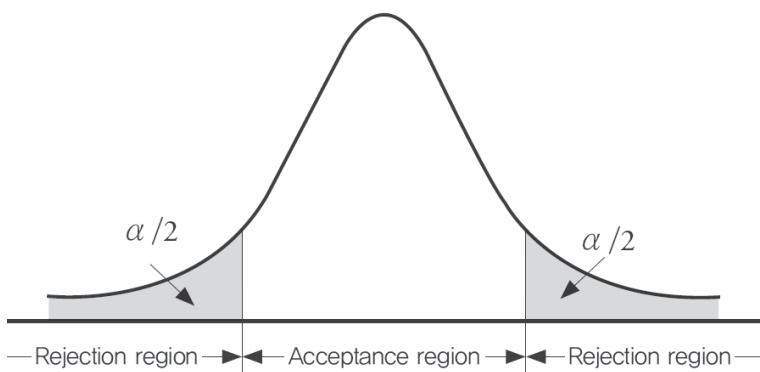


Fig. 3. Regions of Acceptance / Rejection on the Null Hypothesis

Statistical Analysis

Statistical analysis consists of descriptive statistics, where the collected data are summarized into easily-understood figures, and inferential statistics, where a representative sample of the population is selected and the population parameters are inferred based on the characteristic values of the sample.

For the descriptive and inferential statistical analyses conducted in this text, we employ 350,314 online documents that mention cyber bullying, selected from among 1,210,566 documents collected from January 1, 2013 to June 30, 2017 in Korea.

Table 4 outlines the key items considered in this research. The dependent variables of the research data include the attitudes toward cyber bullying (Positive, Negative) and the type of involvement with cyber bullying (Perpetrator, Victim, Bystander, Complex, Non involved). Moreover, a document's spreading number over one week (Onespread) or two weeks (Twospread) was used, in addition to traditional-bullying (Bullying), as the dependent variables of the analysis of means. The independent variables included strain factors, which are the key factors of Agnew's General Strain Theory (GST). Each of the GST factors was calculated based on its occurrence within a document and the daily frequency of the occurrence. The type of online channel (Channel) and the availability of the original document (Account) were also used as independent variables.

Table 4 Key Items in the Research Data File

Item		Variable Name	Content
Dependent variable	Cyber Bullying Emotion	Attitude	0(Neutral+Negative): Negative, 1: Positive
		Positive	0: No, 1: Yes
		Negative	0: No, 1: Yes
	Cyber Bullying Type	Type	1: Perpetrator, 2: Victim, 3: Bystander 4: Complex 5: Non involved
		Perpetrator	0: No, 1: Yes
		Victim	0: No, 1: Yes
		Bystander	0: No, 1: Yes
		Complex	0: No, 1: Yes
		Non involved	0: No, 1: Yes
	Spreading number	Onespread	Real number
		Twospread	Real number

	Traditional Bullying	Bullying	Real number
Independent variable	GST factors	Strain	0: No, 1: Yes
		Physical	0: No, 1: Yes
		Victim psychology	0: No, 1: Yes
		Self control	0: No, 1: Yes
		Attachment	0: No, 1: Yes
		Passion	0: No, 1: Yes
		Offender psychology	0: No, 1: Yes
		Delinquency	0: No, 1: Yes
		Strain_N	GST Factor (1: one, 2: two over)
	Channel, Account factors	Channel	1: Blog, 2: Board, 3: Cafe, 4: News, 5: Twitter
		Account	0: First, 1: Spread

Descriptive Statistical Analysis

Prior to conduct the various statistical analyses, it is necessary to ascertain the distribution characteristics of the observed variables. The descriptive statistics are used to summarize and review the collected data so as to understand the data's characteristics. Through this process, it is possible to characterize the data variables' distributions; e.g., the central tendency (representative value), dispersion, skewness, and kurtosis.

1) Central tendency (Representative Value)

The central tendency indicates where most of the data is distributed. Because it represents the distribution of a group, it is also referred to as the representative value.

Central tendency	Description
Mean	<ul style="list-style-type: none"> - Known as the average, this is the most commonly used measure of the central tendency. - Population mean (μ) = $\frac{1}{N}(X_1 + X_2 + \dots + X_n) = \frac{1}{N} \sum X_i$ - Sample mean (\bar{x}) = $\frac{1}{n}(X_1 + X_2 + \dots + X_n) = \frac{1}{n} \sum X_i$

Median	<ul style="list-style-type: none"> - The observation located at the halfway position, after the observations have been sorted by magnitude. - For odd n: the $\frac{n+1}{2}$th observation - For even n: the arithmetic mean of the $\frac{n}{2}$th and $\frac{n+1}{2}$th observations.
Mode	<ul style="list-style-type: none"> - The most frequently occurring observation in the data.
Quartiles	<ul style="list-style-type: none"> - After the observations have been sorted by magnitude, the first quartile is the observation located at the 1/4 position, the second quartile is the observation at the 2/4 position, and the third quartile is the observation at the 3/4 position.
Percentiles	<ul style="list-style-type: none"> - After the observations have been sorted by magnitude and divided into 100 even segments, the percentiles are the numbers between each segment. The median is equal to the 50th percentile.

2) Dispersion

Locating the central tendency is insufficient for ascertaining the data distribution. Measures of dispersion aid the understanding of a distribution's characteristics, by indicating the extent of 'spread' in the model of distribution.

Dispersion	Description
Range	<ul style="list-style-type: none"> - After the observations have been sorted by magnitude, this is the difference between the maximum and minimum observations.
Mean deviation	<ul style="list-style-type: none"> - Deviation refers to the distance between an observation and the mean. The mean deviation is the mean of all the absolute deviations. - $MD = \frac{1}{N} \sum X_i - \bar{X}$

Variance and Standard deviation	<ul style="list-style-type: none"> - Variance and standard deviation, the most commonly used measures of dispersion, are essential concepts in statistical analysis. - Population variance: $\sigma^2 = \frac{1}{N} \sum (X_i - \mu)^2$ - Population standard deviation: $\sigma = \sqrt{\frac{1}{N} \sum (X_i - \mu)^2}$ - Sample variance: $s^2 = \frac{1}{n-1} \sum (X_i - \bar{X})^2$ - Sample standard deviation: $s = \sqrt{\frac{1}{n-1} \sum (X_i - \bar{X})^2}$ <p>X: observations, N: no. of X, μ: population mean, \bar{X}: sample mean</p>
Coefficient of variance	<ul style="list-style-type: none"> - A measure used to easily understand the relative extent of dispersion - Coefficient of variance (CV) = $\frac{s}{\bar{X}}$ or $\frac{s}{\bar{X}} \times 100$ <p>s: standard deviation, \bar{X}: mean</p>
Skewness and Kurtosis	<ul style="list-style-type: none"> - Skewness refers to the extent to which the distribution's shape is 'swept' to the left or right of the central position. If the central position occurs to the left, the skewness is positive (+). If it occurs to the right, the skewness is negative (-). - Kurtosis refers to the extent of a distribution's 'pointed'. A kurtosis value of 3 indicates that the distribution is similar to the normal distribution, while smaller values indicate less pointed and higher values indicate more pointed.

① Analysis of Central tendency and Dispersion

Step 1: Install the package needed to analyze the central tendency and dispersion.

```
> install.packages('Rcmdr')
- Install the R Commander package, which supports a graphic user interface environment.
```

> library(Rcmdr): Load the Rcmdr package.

- Because we only use R Commander functions in this study, we do not use the R Commander menus. Therefore, click the minimize button on the R Commander box and send it to the Windows taskbar.

Step 2 : Analyze the central tendency and dispersion

```
> setwd("c:/cyber_bullying_methodology"): Set the working directory.
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)
```

- Load the data file and assign it to cyber_bullying.

```
> attach(cyber_bullying)
    - Attach the cyber_bullying data frame as execution data.
```

> length(Strain): Compute the sample size of Strain.

> mean(Strain): Compute the daily mean of Strain.

> var(Strain): Compute the daily variance of Strain.

> var(Strain)*(length(Strain)-1)/length(Strain)
 - Compute the daily population variance [sample variance*(n-1)/n] of Strain.

> sd(Strain): Compute the daily standard deviation of Strain.

> sd(Strain)*(length(Strain)-1)/length(Strain)
 - Compute the daily population standard deviation [sample standard deviation*(n-1)/n] of Strain.

> sd(Strain)/mean(Strain)
 - Compute the coefficient of variation (CV) of Strain.

> sd(Delinquency)/mean(Delinquency)
 - Compute the coefficient of variation (CV) of Delinquency.

> quantile(Strain): Compute the quartiles of Strain.

> quantile(Delinquency): Compute the quartiles of Delinquency.

The screenshot shows the R Commander interface. On the left, the 'R Script' tab is active, displaying R code. On the right, the 'R Console' tab is active, displaying the output of the R code. The R code includes commands for installing packages, reading a data file, attaching the data frame, and performing various statistical calculations like mean, variance, standard deviation, and coefficient of variation. It also shows the computation of quartiles for both Strain and Delinquency.

```

> install.packages('Rcmdr')
Warning: package 'Rcmdr' is in use and will not be installed
> library(Rcmdr)
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)
> attach(cyber_bullying)
The following objects are masked from cyber_bullying (pos = 3):
Attachment, Delinquency, Offender_psychology, Passion, Physical,
Self_control, Strain, Victim_psychology, fear

> length(Strain)
[1] 365
> mean(Strain)
[1] 41.32877
> var(Strain)
[1] 716.7982
> var(Strain)*(length(Strain)-1)/length(Strain)
[1] 714.8344
> sd(Strain)
[1] 26.77309
> sd(Strain)*(length(Strain)-1)/length(Strain)
[1] 26.69974
> sd(Strain)/mean(Strain)
[1] 0.6478076
> sd(Delinquency)/mean(Delinquency)
[1] 0.6940196
> quantile(Strain)
 0% 25% 50% 75% 100%
 14   28   36   47  376
> quantile(Delinquency)
 0% 25% 50% 75% 100%
 7   17   24   31  259
>

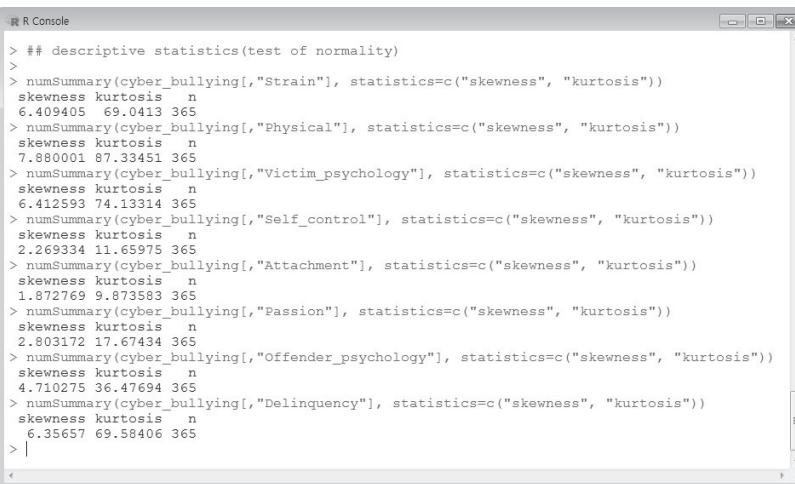
```

Messages

```
[1] "R CMDR R Commander Version 2.5-1 - www.r-project.org 24/02/2008 2010"
[2] "WARNING: The Windows version of the R Commander works best under GUI with the single-document interface (SDI); see ?Commander."
```

Step 3: Conduct normality tests of the variables.

```
> numSummary(cyber_bullying[, "Strain"], statistics=c("skewness",
  "kurtosis")): Test the normality of Strain.
> numSummary(cyber_bullying[, "Physical"], statistics=c("skewness",
  "kurtosis"))
> numSummary(cyber_bullying[, "Victim_psychology"], statistics=c
  ("skewness", "kurtosis"))
> numSummary(cyber_bullying[, "Self_control"], statistics=c("skewness",
  "kurtosis"))
> numSummary(cyber_bullying[, "Attachment"], statistics=c("skewness",
  "kurtosis"))
> numSummary(cyber_bullying[, "Passion"], statistics=c("skewness",
  "kurtosis"))
> numSummary(cyber_bullying[, "Offender_psychology"], statistics=c
  ("skewness", "kurtosis"))
> numSummary(cyber_bullying[, "Delinquency"], statistics=c("skewness",
  "kurtosis"))
```

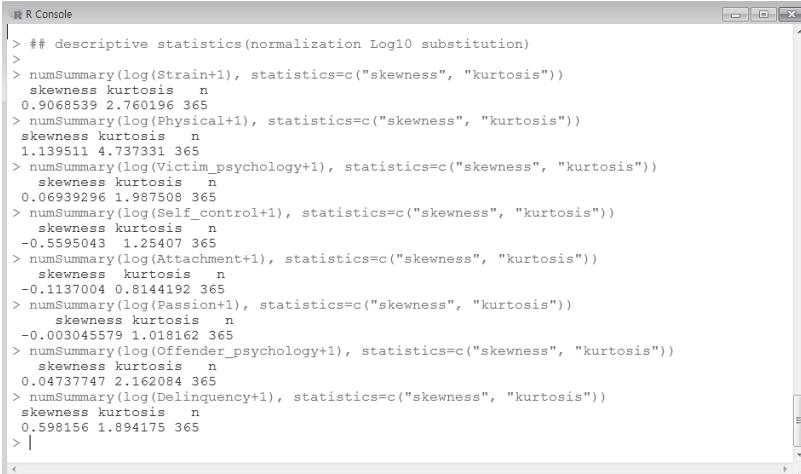


The screenshot shows the R Console window with the following text:

```
R Console
> ## descriptive statistics(test of normality)
>
> numSummary(cyber_bullying[, "Strain"], statistics=c("skewness", "kurtosis"))
skewness kurtosis n
6.409405 69.0413 365
> numSummary(cyber_bullying[, "Physical"], statistics=c("skewness", "kurtosis"))
skewness kurtosis n
7.880001 87.33451 365
> numSummary(cyber_bullying[, "Victim_psychology"], statistics=c("skewness", "kurtosis"))
skewness kurtosis n
6.412593 74.13314 365
> numSummary(cyber_bullying[, "Self_control"], statistics=c("skewness", "kurtosis"))
skewness kurtosis n
2.269334 11.65975 365
> numSummary(cyber_bullying[, "Attachment"], statistics=c("skewness", "kurtosis"))
skewness kurtosis n
1.872769 9.873583 365
> numSummary(cyber_bullying[, "Passion"], statistics=c("skewness", "kurtosis"))
skewness kurtosis n
2.803172 17.67434 365
> numSummary(cyber_bullying[, "Offender_psychology"], statistics=c("skewness", "kurtosis"))
skewness kurtosis n
4.710275 36.47694 365
> numSummary(cyber_bullying[, "Delinquency"], statistics=c("skewness", "kurtosis"))
skewness kurtosis n
6.35657 69.58406 365
> |
```

Interpretation: The normality assumption is satisfied if the absolute values of the skewness and kurtosis are less than 3 and 10, respectively (Kline, 2010). Because all of the GST factors except for Attachment do not satisfy the normality assumption, we test for normality after taking the common logarithm of every factor.

```
> numSummary(log(Strain+1), statistics=c("skewness", "kurtosis"))
  - Test the normality of Strain after taking its common logarithm.
  - '+1': a method for avoiding undefined values [log(0)] when taking
    logarithms of variables.
> numSummary(log(Physical+1), statistics=c("skewness", "kurtosis"))
> numSummary(log(Victim_psychology+1), statistics=c("skewness",
  "kurtosis"))
> numSummary(log(Self_control+1), statistics=c("skewness", "kurtosis"))
> numSummary(log(Attachment+1), statistics=c("skewness", "kurtosis"))
> numSummary(log(Passion+1), statistics=c("skewness", "kurtosis"))
> numSummary(log(Offender_psychology+1), statistics=c("skewness",
  "kurtosis"))
> numSummary(log(Delinquency+1), statistics=c("skewness", "kurtosis"))
```



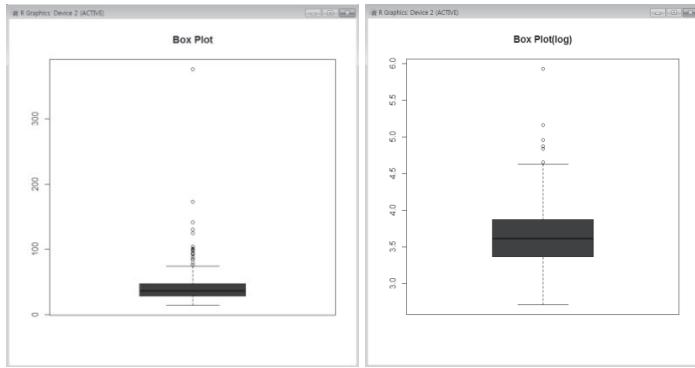
The screenshot shows the R Console window with the following R code and its output:

```
> ## descriptive statistics(normalization Log10 substitution)
>
> numSummary(log(Strain+1), statistics=c("skewness", "kurtosis"))
skewness kurtosis   n
0.9068539 2.760196 365
> numSummary(log(Physical+1), statistics=c("skewness", "kurtosis"))
skewness kurtosis   n
1.139511 4.737331 365
> numSummary(log(Victim_psychology+1), statistics=c("skewness", "kurtosis"))
skewness kurtosis   n
0.06939296 1.987508 365
> numSummary(log(Self_control+1), statistics=c("skewness", "kurtosis"))
skewness kurtosis   n
-0.5595043 1.25407 365
> numSummary(log(Attachment+1), statistics=c("skewness", "kurtosis"))
skewness kurtosis   n
-0.1137004 0.8144192 365
> numSummary(log(Passion+1), statistics=c("skewness", "kurtosis"))
skewness kurtosis   n
-0.003045579 1.018162 365
> numSummary(log(Offender_psychology+1), statistics=c("skewness", "kurtosis"))
skewness kurtosis   n
0.04737747 2.162084 365
> numSummary(log(Delinquency+1), statistics=c("skewness", "kurtosis"))
skewness kurtosis   n
0.598156 1.094175 365
>
```

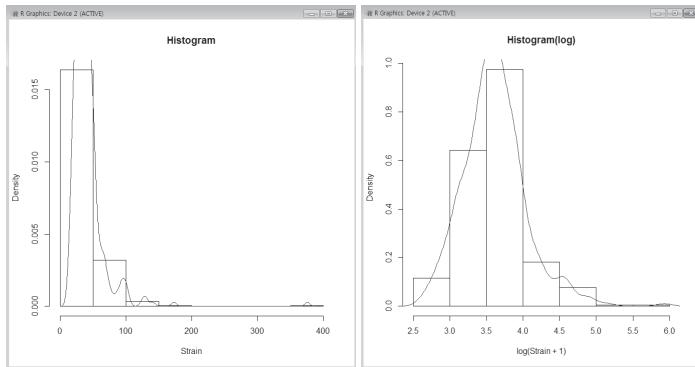
Interpretation: Testing for normality after taking the common logarithm of all GST factors, the normality assumption is found to be satisfied.

■ Visualization of Continuous Variables (boxplot, histogram, line)

```
> boxplot(Strain, col='blue', main='Box Plot')
> boxplot(log(Strain+1), col='blue', main='Box Plot(log)')
  - After the log transformation, Strain is normalized and has almost no
    outliers.
```



```
> hist(Strain, prob=T, main='Histogram'): Plot a histogram of Strain.  
> lines(density(Strain), col='blue')  
  - Add a fitted density curve to the histogram.  
> hist(log(Strain+1), prob=T, main='Histogram(log)')  
  - Plot a histogram of the log-transformed Strain.  
> lines(density(log(Strain+.1)), col='blue')  
  - Add a fitted density curve to the histogram.  
  - After the log transformation, the Strain factor follows a normal  
    distribution.
```

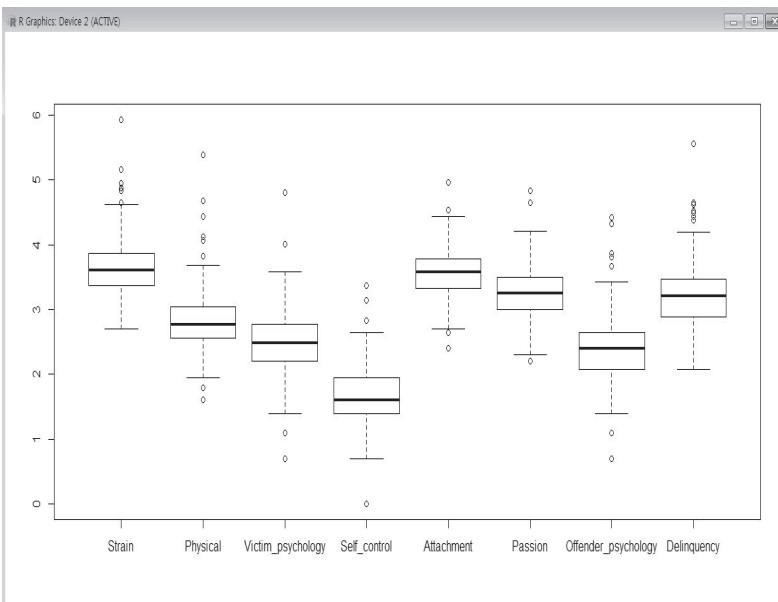


```
> cyber_bullying_s = data.frame(Strain, Physical, Victim_psychology, Self_control, Attachment, Passion, Offender_psychology, Delinquency)  
  - Assign the eight vectors of the GST factors to the cyber_bullying_s  
    data frame object.  
> boxplot(cyber_bullying_s)  
  - Plot boxplots for all factors in the cyber_bullying_s data frame.
```

```
> cyber_bullying_l=log(cyber_bullying_s+1)
  - Log-transform all factors and assign them to the cyber_bullying_l
    object.
> boxplot(cyber_bullying_l)
  - Plot boxplots for all factors in the cyber_bullying_l object.
```

R Console window showing R code for descriptive statistics visualization:

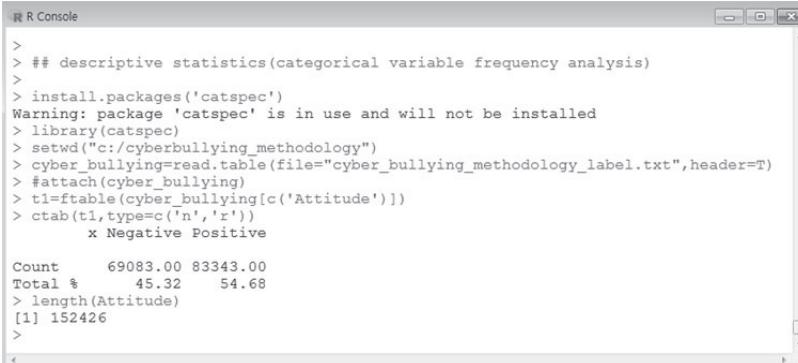
```
> ## descriptive statistics visualization(boxplot, histogram, line)
>
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)
>
> boxplot(Strain, col='blue', main='Box Plot')
> boxplot(log(Strain+1), col='blue', main='Box Plot(log)')
> hist(Strain, prob=T,main='Histogram' )
> lines(density(Strain), col='blue')
> hist(log(Strain+1), prob=T,main='Histogram(log)' )
> lines(density(log(Strain+1)), col='blue')
> cyber_bullying_s=data.frame(Strain,Physical,Victim_psychology,Self_control,
+ Attachment,Passion,Offender_psychology,Delinquency)
> boxplot(cyber_bullying_s)
> cyber_bullying_l=log(cyber_bullying_s+1)
> boxplot(cyber_bullying_l)
>
> |
```



② Frequency Analysis of Categorical Variables

Because the concepts of mean and standard deviation are inapplicable to categorical variables, it is necessary to compute the frequencies and ratios of the values. Therefore, in the case of categorical variables, it is meaningful to examine the distributions through characteristics such as the frequency, median, mode, range, and percentiles.

```
> install.packages('catspec')
  - Install a package that supports contingency tables.
> library(catspec): Load the catspec package.
> setwd("c:/cyberbullying_methodology"): Set the working directory.
> cyber_bullying=read.table(file="cyber_bullying_methodology_label.txt",
  header=T)
  - Load the data file and assign it to the cyber_bullying object.
> attach(cyber_bullying): Attach 'cyber_bullying' as execution data.
> t1=ftable(cyber_bullying[c('Attitude')])
  - 'ftable' is a function for creating 'flat' contingency tables.
  - Conduct frequency analysis of 'Attitude', and assign the contingency
    table to t1.
> ctab(t1,type=c('n','r'))
  - Display the frequencies and the frequency percentages of 'Attitude'
    to the screen.
> length(Attitude): Display the total frequency of 'Attitude' to the screen.
```



The screenshot shows the R Console window with the following R code and its output:

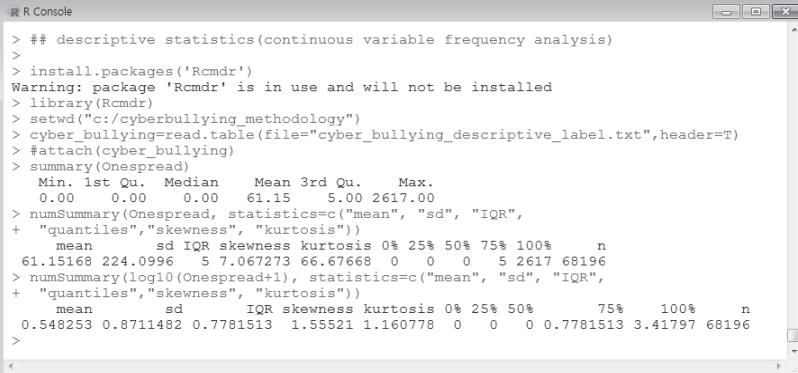
```
>
> ## descriptive statistics(categorical variable frequency analysis)
>
> install.packages('catspec')
Warning: package 'catspec' is in use and will not be installed
> library(catspec)
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_methodology_label.txt",header=T)
> #attach(cyber_bullying)
> t1=ftable(cyber_bullying[c('Attitude')])
> ctab(t1,type=c('n','r'))
  x Negative Positive
Count      69083.00 83343.00
Total %     45.32    54.68
> length(Attitude)
[1] 152426
>
```

Interpretation: Out of a total of 152,426 online documents, 45.32% (69,083 cases) showed negative attitudes toward cyber bullying, while 54.68% (83,343 cases) showed positive attitudes.

③ Frequency Analysis of Continuous Variables

In the case of continuous variables, the mean and variance are used to determine the extent of the dispersion, and the measures of skewness and kurtosis are used to determine normality. Absolute values of skewness and kurtosis smaller than 3 and 10, respectively, indicate that the normality assumption is satisfied (Kline, 2010).

```
> install.packages('Rcmdr')
> library(Rcmdr)
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",
  header=T)
> attach(cyber_bullying)
> summary(Onespread)
  - Compute basic descriptive statistics for Onespread.
> numSummary(Onespread, statistics=c("mean", "sd", "IQR",
  "quantiles", "skewness", "kurtosis"))
  - Compute designated descriptive statistics for 'Onespread'.
> numSummary(log10(Onespread+1), statistics=c("mean", "sd", "IQR",
  "quantiles", "skewness", "kurtosis"))
  - Compute designated descriptive statistics for 'Onespread' after taking
    its logarithm.
```



The screenshot shows the R Console window with the following output:

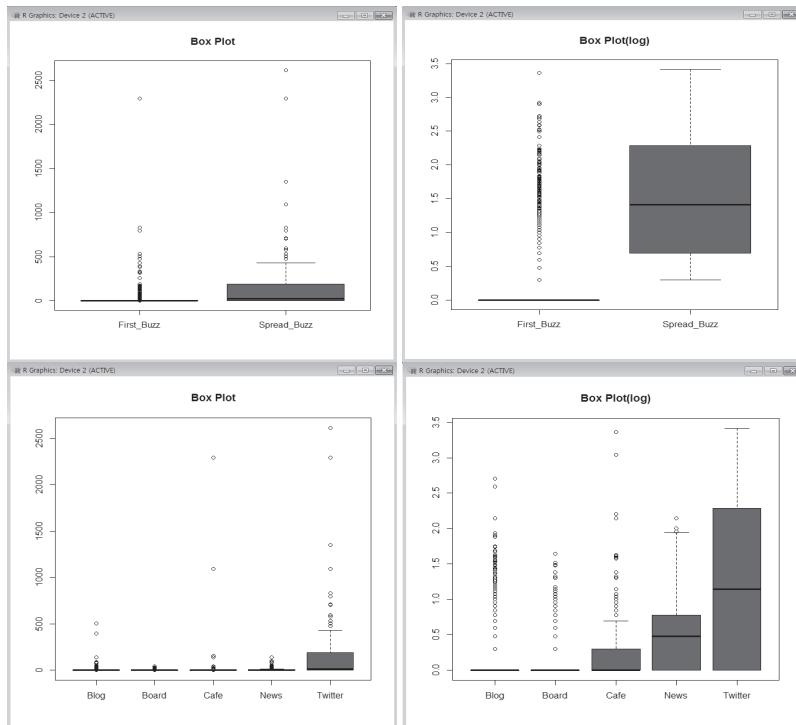
```
> ## descriptive statistics(continuous variable frequency analysis)
>
> install.packages('Rcmdr')
Warning: package 'Rcmdr' is in use and will not be installed
> library(Rcmdr)
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)
> #attach(cyber_bullying)
> summary(Onespread)
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  0.00   0.00   0.00  61.15   5.00 2617.00
> numSummary(Onespread, statistics=c("mean", "sd", "IQR",
+ "quantiles", "skewness", "kurtosis"))
  mean      sd IQR skewness kurtosis 0% 25% 50% 75% 100% n
  61.15168 224.09965 7.0672733 66.67668 0 0 0 5 2617 68196
> numSummary(log10(Onespread+1), statistics=c("mean", "sd", "IQR",
+ "quantiles", "skewness", "kurtosis"))
  mean      sd IQR skewness kurtosis 0% 25% 50% 75% 100% n
  0.548253 0.8711482 0.7781513 1.55521 1.160778 0 0 0 0.7781513 3.41797 68196
>
```

Interpretation: Analysis of 68,196 online documents indicates that the one-week spread of documents (Onespread) has a mean of 61.15, a standard deviation (distance from mean) of 224.1, a central position that is located to the left of the distribution (skewness: 7.07), and a distribution

that is pointier than the normal distribution (kurtosis: 66.78). As Onespread did not satisfy the normality assumption, we repeated the normality test after taking its common logarithm. The log-transformed Onespread has an absolute skewness value (1.56) smaller than 3, and an absolute kurtosis value (1.16) smaller than 10, thus satisfying the normality assumption.

■ Visualization of Continuous Variables

```
> boxplot(Onespread~Account,col='red', main='Box Plot')
  – Create boxplots of Onespread using Account.
> boxplot(log10(Onespread+1)~Account,col='red', main='Box Plot(log)')
  - Create boxplots of log-transformed Onespread using Account.
> boxplot(Onespread~Channel,col='red', main='Box Plot')
  - Create boxplots of log-transformed Onespread using Channel.
> boxplot(log10(Onespread+1)~Channel,col='red', main='Box Plot(log)')
```



Inferential Statistical Analysis

Inferential statistics is a statistical analysis method for estimating population parameters using sample statistics. The purpose is to determine whether the analysis conducted on the sample can be generalized to the population. In inferential statistics, sample statistics are used to estimate population parameters, by means of hypothesis testing. A means analysis is used to estimate the population means depending on the scale properties (continuous or categorical) of the dependent and independent variables. Cross-tabulation analysis, correlation analysis, factor analysis, and cluster analysis are used to determine the associations between variables, while regression and logistic regression analyses are used to examine the dependencies between variables.

④ Cross-Tabulation Analysis

While frequency analysis is a descriptive statistical method for analyzing the characteristics of a single variable, cross-tabulation analysis is a inferential statistical method that can be used to analyze the correlation between two or more variables. As opposed to frequency analysis, where a frequency table is created for a single variable, cross-tabulation analysis uses tables with two or more rows and columns to examine links between variables.

Since collected data are always a part of a sample that has been extracted from the population, and because the parameters that determine a population's characteristics are usually unknown, the population parameters are estimated based on the statistics computed from the observable sample.

In this regard, the χ^2 -test employs a contingency table of rows and columns with statistics for testing for the independence and homogeneity between two variables. The researcher can then determine whether the relationship observed between the two variables in the sample also hold in the population.

■ Independence test

A test performed on a sample extracted from a population where the observation subjects have not been determined beforehand, as is the case in most statistical survey.

■ Homogeneity test

A test performed on a sample extracted from a population where the observation subjects have been determined beforehand. This is usually employed in the analysis of clinical trial results (e.g., comparing a group that has been treated with vitamin C vs. the control group that has not been treated).

■ χ^2 -test procedure

Step 1: Set the null hypothesis (H_0); the two variables are independent of each other.

Step 2: Set the level of significance (α): (.001,.01,.05,.1).

Step 3: Using the sample statistics, compute the p-value.

Step 4: If $p < \alpha$, reject the null hypothesis and accept the alternative hypothesis.

■ Association measures

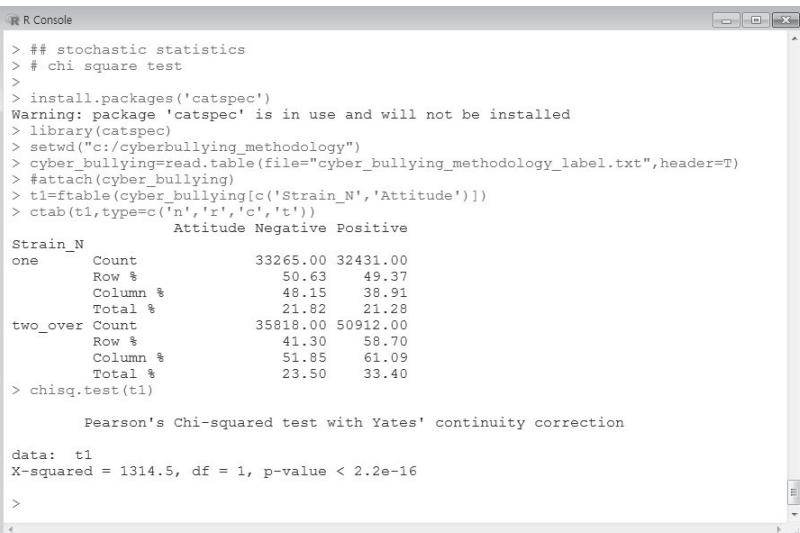
- If the χ^2 -test result rejects H_0 , these measures indicate the extent of association between two variables.
- The contingency coefficient is used when the number of rows equals the number of columns ($0 \leq C \leq 1$).
- Cramer's V can be used when the number of rows does not equal the number of columns ($0 \leq V \leq 1$).
- Kendall's τ (Tau) is used with ordinal data where the number of rows equals the number of columns (τ_b), or where the number of rows does not equal the number of columns (τ_c).
- Somers' D is used with ordinal data where the causal link between two variables is predetermined; e.g., academic major vs. occupation after graduation. ($-1 \leq D \leq 1$).
- η (Eta) indicates the degree of association between categorical data and continuous data. ($0 \leq \eta \leq 1$, values closer to 1 indicate a stronger association).
- The Pearson's R correlation coefficient indicates the extent of the linear association between interval data ($-1 \leq R \leq 1$).

Research Problem: Conduct a cross-tabulation analysis (χ^2 -test) between attitudes toward cyber bullying (Attitude) and a GST factor (Strain N).

```

> install.packages('catspec')
  - Install a package that supports contingency tables.
> library(catspec): Load the catspec package.
> setwd("c:/cyberbullying_methodology"): Set the working directory.
> cyber_bullying=read.table(file="cyber_bullying_methodology_label.txt",
  header=T)
  - Load the data file and assign it to the cyber_bullying object.
> attach(cyber_bullying)
  - Attach 'cyber_bullying' as execution data.
> t1=ftable(cyber_bullying[c('Strain_N','Attitude')])
  - 'ftable' is a function for creating 'flat' contingency tables.
  - Conduct cross-tabulation ('Strain_N', 'Attitude'); then, assign the
    contingency table to t1.
> ctab(t1,type=c('n','r','c','t'))
  - Display the frequency, row, column, and total % of the two-way
    contingency table on the screen.
> chisq.test(t1)
  - Display the test statistic of the  $\chi^2$ -test for the two-way contingency
    table on the screen.

```



The screenshot shows the R Console window with the following content:

```

R Console

> ## stochastic statistics
> # chi square test
>
> install.packages('catspec')
Warning: package 'catspec' is in use and will not be installed
> library(catspec)
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_methodology_label.txt",header=T)
> #attach(cyber_bullying)
> t1=ftable(cyber_bullying[c('Strain_N','Attitude')])
> ctab(t1,type=c('n','r','c','t'))
      Attitude Negative Positive
Strain_N
one      Count          33265.00 32431.00
         Row %           50.63   49.37
         Column %        48.15   38.91
         Total %          21.82   21.28
two_over Count          35818.00 50912.00
         Row %           41.30   58.70
         Column %        51.85   61.09
         Total %          23.50   33.40
> chisq.test(t1)

Pearson's Chi-squared test with Yates' continuity correction

data: t1
X-squared = 1314.5, df = 1, p-value < 2.2e-16
>

```

Interpretation: With only one GST factor (Strain_N), negative attitudes to cyber bullying were more frequent (50.63%, 33,265 cases). In the case of two or more GST factors, positive attitudes to cyber bullying were more

frequent (58.70%, 50,912 cases). The χ^2 -test result indicates that there is a significant difference between the two variables ($\chi^2 = 1314.5, p(2.2 \times 10^{-16}) < .001$).

- Conduct cross-tabulation analysis in SPSS format using the “gmodels” package

```
> install.packages('gmodels')
> library(gmodels)
> CrossTable(cyber_bullying$Strain_N, cyber_bullying$Attitude,
+ expected=T,format='SPSS')
- Run cross-tabulation analysis in SPSS format
```

The screenshot shows the R Console window with the following output:

```
R R Console

> CrossTable(cyber_bullying$Strain_N, cyber_bullying$Attitude,
+ expected=T,format='SPSS')

Cell Contents
|-----|
|      Count |
|      Expected Values |
| Chi-square contribution |
|      Row Percent |
|      Column Percent |
|      Total Percent |
|-----|

Total Observations in Table:  152426

              |   cyber_bullying$Attitude
cyber_bullying$Strain_N | Negative | Positive | Row Total |
|-----|-----|-----|-----|
one | 33265 | 32431 | 65696 |
| 29774.952 | 35921.048 | |
| 409.083 | 339.089 | |
| 50.635% | 49.365% | 43.100% |
| 48.152% | 38.913% | |
| 21.824% | 21.277% | |
|-----|-----|-----|-----|
two_over | 35818 | 50912 | 86730 |
| 39308.048 | 47421.952 | |
| 309.871 | 256.852 | |
| 41.298% | 58.702% | 56.900% |
| 51.848% | 61.087% | |
| 23.499% | 33.401% | |
|-----|-----|-----|-----|
Column Total | 69083 | 83343 | 152426
| 45.322% | 54.678% | |

Statistics for All Table Factors

Pearson's Chi-squared test
-----
Chi^2 = 1314.896    d.f. = 1    p =  6.547576e-288

Pearson's Chi-squared test with Yates' continuity correction
-----
Chi^2 = 1314.519    d.f. = 1    p =  7.905845e-288
```

■ Analyze the association measures

```
> cyber_bullying=read.table(file="cyber_bullying_methodology_numeric.txt", header=T)
> with(cyber_bullying, cor.test(Strain_N,Attitude,method='pearson'))
  - Compute the Pearson correlation coefficient.
> with(cyber_bullying, cor.test(Strain_N,Attitude,method='kendall'))
  - Compute Kendall's  $\tau$  (tau).
> cv.test = function(x,y) {
  CV = sqrt(chisq.test(x, y, correct=FALSE)$statistic /
  (length(x) * (min(length(unique(x)),length(unique(y))) - 1)))
  print.noquote("Cramer V / Phi:")
  return(as.numeric(CV))
}
- Create a function (cv.test) that computes Cramer's V.
> with(cyber_bullying, cv.test(Strain_N,Attitude)): Compute Cramer's V.
```



The screenshot shows the R Console window with the following output:

```
R Console
> ## measures of association
>
> cyber_bullying=read.table(file="cyber_bullying_methodology_numeric.txt", header=T)
> with(cyber_bullying, cor.test(Strain_N,Attitude,method='pearson'))

Pearson's product-moment correlation

data: Strain_N and Attitude
t = 36.419, df = 152420, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.08789952 0.09785327
sample estimates:
cor
0.09287872

> with(cyber_bullying, cor.test(Strain_N,Attitude,method='kendall'))

Kendall's rank correlation tau

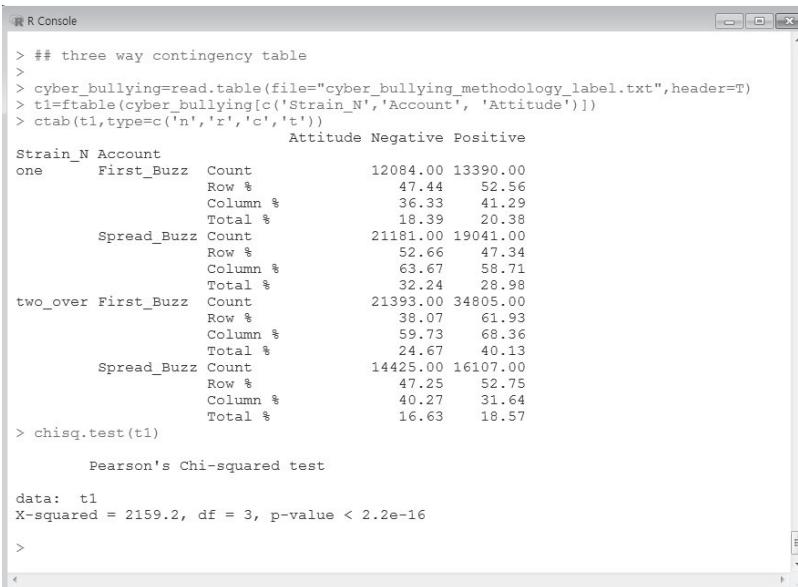
data: Strain_N and Attitude
z = 36.261, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
0.09287872

> ## cramer's v (https://www.r-bloggers.com/example-8-39-calculating-cramers-v/)
> cv.test = function(x,y) {
+   CV = sqrt(chisq.test(x, y, correct=FALSE)$statistic /
+   (length(x) * (min(length(unique(x)),length(unique(y))) - 1)))
+   print.noquote("Cramer V / Phi:")
+   return(as.numeric(CV))
+ }
>
> ## we can get CramerV as
> with(cyber_bullying, cv.test(Strain_N,Attitude))
[1] Cramer V / Phi:
[1] 0.09287872
> |
```

Interpretation: Cramer's V association measure between Strain_N and Attitude is found to be 0.0929.

■ Analysis of Three-way Contingency Tables

```
> cyber_bullying=read.table(file="cyber_bullying_methodology_label.txt",
  header=T)
> t1=ftable(cyber_bullying[c('Strain_N','Account', 'Attitude')])
  - Assign the values of the three-way contingency table to t1.
> ctab(t1,type=c('n','r','c','t'))
  - Print the frequency, row, column, and total % of the three-way
  contingency table on the screen.
> chisq.test(t1)
  - Print the test statistic of the  $\chi^2$ -test for the three-way contingency
  table on the screen.
```



The screenshot shows the R Console window with the following output:

```
R Console

> ## three way contingency table
>
> cyber_bullying=read.table(file="cyber_bullying_methodology_label.txt",header=T)
> t1=ftable(cyber_bullying[c('Strain_N','Account', 'Attitude')])
> ctab(t1,type=c('n','r','c','t'))
      Attitude Negative Positive
Strain_N Account
one      First_Buzz Count      12084.00 13390.00
          Row %       47.44   52.56
          Column %     36.33   41.29
          Total %      18.39   20.38
          Spread_Buzz Count      21181.00 19041.00
          Row %       52.66   47.34
          Column %     63.67   58.71
          Total %      32.24   28.98
two_over First_Buzz Count      21393.00 34805.00
          Row %       38.07   61.93
          Column %     59.73   68.36
          Total %      24.67   40.13
          Spread_Buzz Count      14425.00 16107.00
          Row %       47.25   52.75
          Column %     40.27   31.64
          Total %      16.63   18.57
> chisq.test(t1)

Pearson's Chi-squared test

data: t1
X-squared = 2159.2, df = 3, p-value < 2.2e-16
>
```

⑤ Test of Means (One-Sample T-test)

The one-sample T-test is a method of testing whether the mean of a single sample is different from the population mean, where the population mean is known.

Research Hypothesis: ($H_0: \mu_1 = 100$, $H_1: \mu_1 \neq 100$). The test concerns whether the mean of the one-week spread of cyber bullying (Onespread) is different from the mean of the one-week spread in the population, which is equal to 100. In a previous study(song et al., 2016), the one-week spread of suicide contemplation was found to be 100 times.

```
> setwd("c:/cyberbullying_methodology"): Set the working directory.
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)
  - Load the data file and assign it to cyber_bullying.
> t.test(cyber_bullying[c('Onespread')],mu=100)
  - Run the one-sample t-test for Onespread.
```

```
R Console
> ## one sample T Test
>
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)
> t.test(cyber_bullying[c('Onespread')],mu=100)

One Sample t-test

data: cyber_bullying[c("Onespread")]
t = -45.27, df = 68195, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 100
95 percent confidence interval:
59.46971 62.83365
sample estimates:
mean of x
61.15168

> |
```

Interpretation: The mean of the one-week spread, measured based on a sample of 68,196 online documents, was 61.15. This is found to be significantly lower than the test value of 100 ($t = -45.27, p = .000 < .001$). Thus, the alternative hypothesis ($H_1: \mu_1 \neq 100$) is accepted and the confidence interval is found to be 59.47~62.83. As this interval does not include 0, the alternative hypothesis is supported.

⑥ Test of Means (Independent-Sample T-test)

The independent-sample T-test is a method of testing the difference in population means using two samples of size n_1 and n_2 , which have been extracted from two populations. In this test, the means testing is preceded by testing for equal-variance ($H_0: \sigma_1^2 = \sigma_2^2$). If the equal-variance assumption is satisfied, the pooled variance is used to conduct the T-test. If not, Welch's T-test is conducted.

Research Hypothesis: There is a difference between the means of the one-week spread (Onespread) in the two groups (First, Spread) of effective accounts (Account).

```
> rm(list=ls()): Initialize all variables.  
> setwd("c:/cyberbullying_methodology")  
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)  
> var.test(Onespread~Account,cyber_bullying)  
  - Conduct the test for equal-variance.  
  - In this case, the equal-variance assumption is not satisfied ( $F = 0.002$ ,  
  $p < 0.001$ ).  
> t.test(Onespread~Account,cyber_bullying): unequal-variance.  
> t.test(Onespread~Account,var.equal=T,cyber_bullying): equal-variance.
```

The screenshot shows the R Console window with the following output:

```
R Console  
> ## independent sample T Test  
> rm(list=ls())  
> setwd("c:/cyberbullying_methodology")  
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)  
> var.test(Onespread~Account,cyber_bullying)  
  F test to compare two variances  
  
data: Onespread by Account  
F = 0.0015145, num df = 45372, denom df = 22822, p-value < 2.2e-16  
alternative hypothesis: true ratio of variances is not equal to 1  
95 percent confidence interval:  
 0.001480749 0.001548887  
sample estimates:  
ratio of variances  
 0.001514499  
  
>  
> # in case of a different variance(Welch T Test)  
> t.test(Onespread~Account,cyber_bullying)  
  Welch Two Sample t-test  
  
data: Onespread by Account  
t = -76.375, df = 22857, p-value < 2.2e-16  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 -185.5195 -176.2355  
sample estimates:  
mean in group First Buzz mean in group Spread Buzz  
 0.6178124 181.4953337  
  
>  
> # in case of a same variance(pooled variance T Test)  
> t.test(Onespread~Account,var.equal=T,cyber_bullying)  
  Two Sample t-test  
  
data: Onespread by Account  
t = -107.57, df = 68194, p-value < 2.2e-16  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 -184.1733 -177.5817  
sample estimates:  
mean in group First Buzz mean in group Spread Buzz  
 0.6178124 181.4953337  
> |  
>
```

Interpretation: Prior to conducting the independent-sample T-test, it is necessary to first establish the homogeneity (equal variance) between the two groups. Testing for the equal-variance of the one-week spread (Onespread) found that the equal-variance assumption was not satisfied, with the F statistic equal to $F = 0.002$ and the p -value being $p = .000 < .001$. The difference in mean between the two groups (First, Spread) of effective accounts (Account) was found to be significant [$t = -76.375$ ($p < .001$)].

Had the equal-variance assumption been satisfied, the difference in mean would still have been significant [$t = -107.57$ ($p < .001$)].

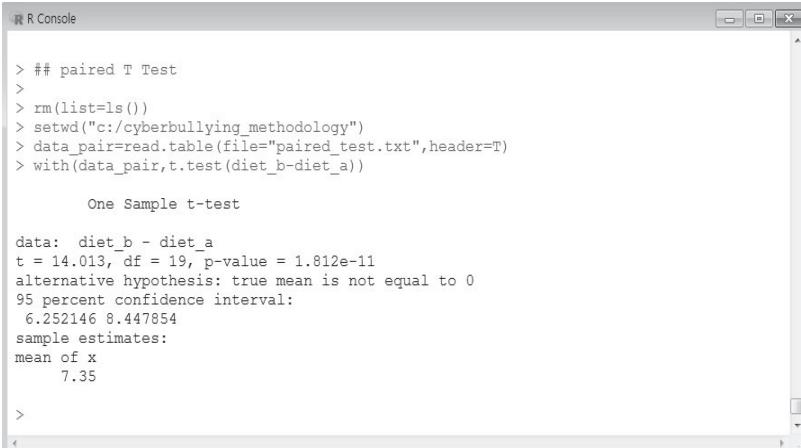
⑦ Test of Means (Paired T-Test)

The paired T-test is a method of testing for the difference in population means when two samples of size n_1 and n_2 have been extracted from the same population.

Research Hypothesis: ($H_0: \mu_1 = \mu_2$, $H_1: \mu_1 \neq \mu_2$). The weights of 20 obese adolescents will be measured before and after they are administered a diet pill, to test whether the pill was effective for weight reduction.

H_0 : The pill is not effective.

```
> rm(list=ls()): Initialize all variables.
> setwd("c:/cyberbullying_methodology"): Set the working directory.
> data_pair=read.table(file="paired_test.txt",header=T)
> with(data_pair,t.test(diet_b-diet_a)): Conduct the paired t-test.
```



The screenshot shows the R Console window with the following output:

```
R Console

> ## paired T Test
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> data_pair=read.table(file="paired_test.txt",header=T)
> with(data_pair,t.test(diet_b-diet_a))

One Sample t-test

data: diet_b - diet_a
t = 14.013, df = 19, p-value = 1.812e-11
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 6.252146 8.447854
sample estimates:
mean of x
 7.35

>
```

Interpretation: A significant difference was found in the mean weights (7.35) of the subjects before and after they were administered a diet pill ($t = 14.013, p < .001$). Thus, the null hypothesis is rejected and the alternative hypothesis is accepted. This supports the claim that the diet pill is effective for weight reduction.

⑧ Test of Means (One-Way ANOVA)

Whereas the T-test is used for examining the difference in means between two groups, for comparisons of means among three or more groups, it is possible to use the F-test based ANOVA (analysis of variance) method. The test admits dependent variables in interval scales or quantitative continuous scales, and the MANOVA (multivariate analysis of variance) may be used in the case of two or more dependent variables.

In particular, one-way ANOVA is used when there is a single independent variable (factor) in a scale with at least three categories. If there are two independent variables, two-way ANOVA may be used. If the null hypothesis $H_0 (\sigma_1^2 = \sigma_2^2 = \dots = \sigma_k^2 = 0)$ in ANOVA is rejected, such that the means across the groups are different, it is necessary to conduct post-hoc analysis (or multiple comparisons) to ascertain whether there are differences in the means of the factors.

Post-hoc analysis methods frequently include Tukey's method (useful for finding significance in small differences), Scheffe's method (useful for finding significance in large differences), and multiple comparisons. Dunnett's multiple comparison method is used when the equal-variance assumption is inapplicable.

Research Hypothesis: ($H_0: \mu_1 - \mu_2 - \dots - \mu_k = 0$, $H_1: \mu_1 - \mu_2 - \dots - \mu_k \neq 0$). That is, H_0 : there are no significant differences in the one-week spread (Onespread) across channels

H_1 : There are significant differences in the one-week spread (Onespread) across channels.

```
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)
> attach(cyber_bullying)
> tapply(Onespread, Channel, mean)
- The tapply() function computes the mean of each group.
> tapply(Onespread, Channel, sd)
- Compute the standard deviation of each group.
```

```
> sel=aov(Onespread~Channel,data=cyber_bullying)
  - Assign the ANOVA table to the sel variable.
> summary(sel): Display the ANOVA table on the screen.
> bartlett.test(Onespread~Channel,data=cyber_bullying)
  - Conduct the test for equal-variance.
```

The screenshot shows the R Console window with the following output:

```
> ## ANOVA(analyses of variance)
> # one way ANOVA(onespread * channel)
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)
> #attach(cyber_bullying)
>
> # calculation of average by group(tapply)
>
> tapply(Onespread, Channel, mean)
   Blog      Board      Cafe      News      Twitter
0.9420638  0.1899315  1.7308564  5.6873034 169.9582487
> tapply(Onespread, Channel, sd)
   Blog      Board      Cafe      News      Twitter
8.426908  1.022174  38.825162 12.015605 351.147912
> sel=aov(Onespread~Channel,data=cyber_bullying)
> summary(sel)
   Df     Sum Sq  Mean Sq F value Pr(>F)
Channel    4 441961155 110490289     2526 <2e-16 ***
Residuals 68191 2982835326        43742
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> 
> # equivalent variance test(bartlett test)
>
> bartlett.test(Onespread~Channel,data=cyber_bullying)

Bartlett test of homogeneity of variances

data: Onespread by Channel
Bartlett's K-squared = 304240, df = 4, p-value < 2.2e-16
>
```

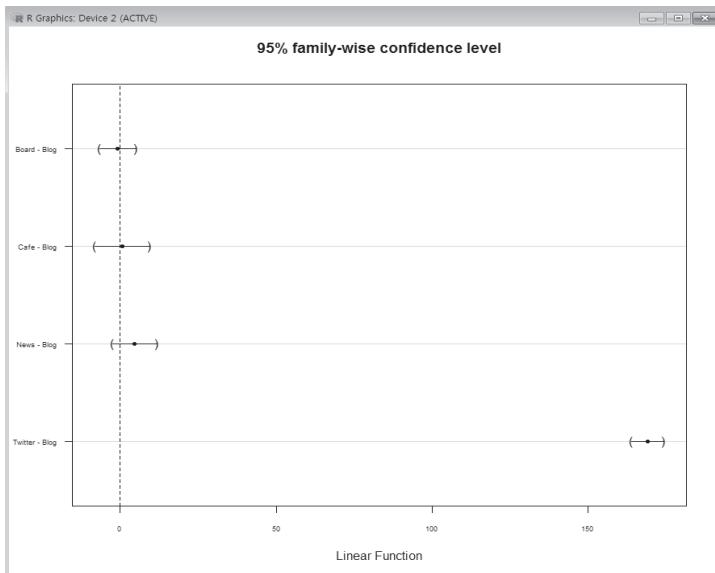
Interpretation: The mean one-week spread (Onespread) was found to be the highest (169.96) in Channel 5 (Twitter). ANOVA results indicate differences in the means of Onespread across different channels ($F = 2,526, p <.001$). The Bartlett test for equal-variance ($B = 304,240, p <.001$) rejected the null hypothesis, indicating that the variances were unequal across the different channels.

■ Conduct the post-hoc analysis (multiple comparisons).

```
> install.packages('multcomp')
  - Install the multiple comparisons package.
> library(multcomp): Load the multiple comparisons package.
> sel=aov(Onespread~Channel,data=cyber_bullying)
  - Assign the ANOVA results to the sel variable.
```

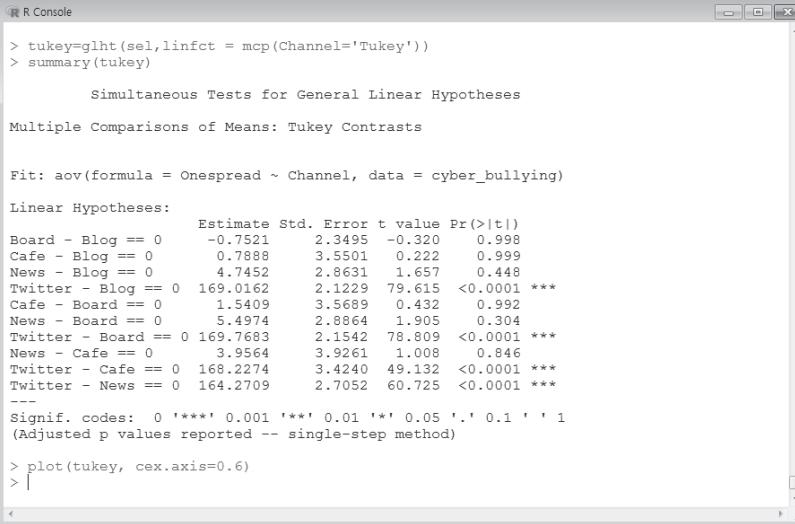
```
> windows(height=5.5, width=5): Set the size of the output screen.  
> dunnett=glht(sel,linfct=mcp(Channel='Dunnett'))  
  - Conduct the Dunnett multiple comparison test.  
> summary(dunnett)  
  - Display the results of the Dunnett multiple comparison test to the screen.  
> plot(dunnett, cex.axis=0.6)  
  - Create a plot of the results, setting the axis label font size to 0.6.
```

```
R Console  
> sel=aov(Onespread~Channel,data=cyber_bullying)  
> windows(height=5.5, width=5)  
>  
> ## equivalent variance(tukey), scheffe), non-equivalent variance(dunnett)  
>  
> dunnett=glht(sel,linfct=mcp(Channel='Dunnett'))  
> summary(dunnett)  
  
Simultaneous Tests for General Linear Hypotheses  
Multiple Comparisons of Means: Dunnett Contrasts  
  
Fit: aov(formula = Onespread ~ Channel, data = cyber_bullying)  
  
Linear Hypotheses:  
Board - Blog == 0 -0.7521 2.3495 -0.320 0.994  
Cafe - Blog == 0 0.7888 3.5501 0.222 0.999  
News - Blog == 0 4.7452 2.8631 1.657 0.300  
Twitter - Blog == 0 169.0162 2.1229 79.615 <0.0001 ***  
---  
Signif. codes: 0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
(Adjusted p values reported -- single-step method)  
> plot(dunnett, cex.axis=0.6)  
> |
```



Interpretation: The Dunnett multiple comparison test results indicate that there are significant mean differences between the Twitter and Blog channels ($p < .001$).

```
> tukey=glht(sel,linfct = mcp(Channel="Tukey"))
  - Run the Tukey multiple comparison.
> summary(tukey)
  - Print the results of the Tukey multiple comparison to the screen.
> summary(tukey)
> plot(tukey, cex.axis=0.6)
  - Create a plot of the Tukey multiple comparison results.
```



The screenshot shows the R Console window with the following output:

```
R Console

> tukey=glht(sel,linfct = mcp(Channel="Tukey"))
> summary(tukey)

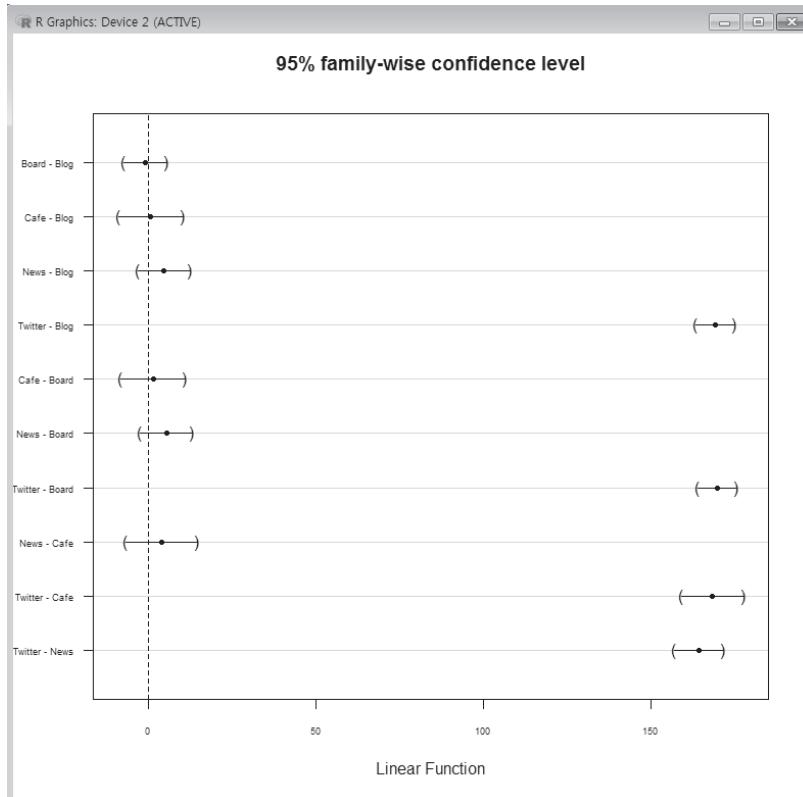
Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = Onespread ~ Channel, data = cyber_bullying)

Linear Hypotheses:
Board - Blog == 0   -0.7521   2.3495  -0.320   0.998
Cafe - Blog == 0    0.7888   3.5501   0.222   0.999
News - Blog == 0    4.7452   2.8631   1.657   0.448
Twitter - Blog == 0 169.0162  2.1229  79.615 <0.0001 ***
Cafe - Board == 0   1.5409   3.5689   0.432   0.992
News - Board == 0   5.4974   2.8864   1.905   0.304
Twitter - Board == 0 169.7683  2.1542  78.809 <0.0001 ***
News - Cafe == 0    3.9564   3.9261   1.008   0.846
Twitter - Cafe == 0 168.2274  3.4240  49.132 <0.0001 ***
Twitter - News == 0 164.2709  2.7052  60.725 <0.0001 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)

> plot(tukey, cex.axis=0.6)
> |
```



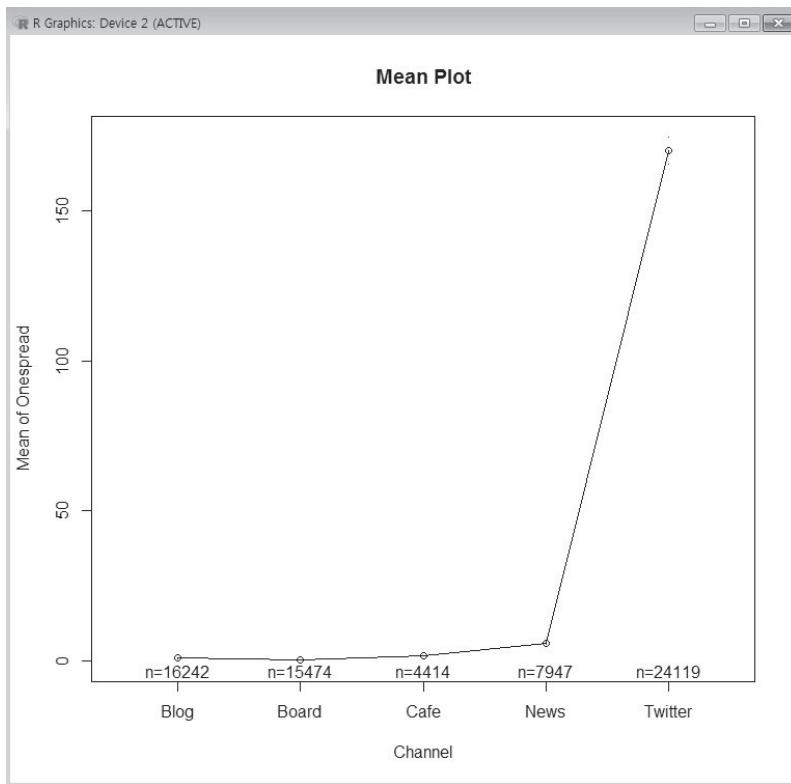
Interpretation: The Tukey multiple comparison grouped Board, Blog, Cafe, and News as the same channel ($p > .1$), and found that the mean in Twitter differed from those of the other channels ($p < .001$). Therefore, the five Channels were split into two groups: (Board, Blog, Cafe, News) and (Twitter).

```
> install.packages('gplots'): Install the gplots package.  

> library(gplots)  

> plotmeans(Onespread~Channel,data=cyber_bullying,xlab='Channel',  

    ylab='Mean of Onespread',main='Mean Plot'): Create a mean plot.
```



Interpretation: As the equal-variance assumption is not satisfied, tests for the groupings can be conducted via the Dunnett multiple comparison or by inspecting the mean plot. According to the mean plot, the one-week spread of cyber bullying can be split into two groups: (Board, Blog, Cafe, News) and (Twitter).

Research Hypothesis: ($H_0: \mu_1 - \mu_2 - \dots - \mu_k = 0$, $H_1: \mu_1 - \mu_2 - \dots - \mu_k \neq 0$).

H_0 : There are no significant differences in the mean Onespread of traditional school bullying across the eight types of involvement in cyber bullying [Type: offender (T1), victim (T2), bystander (T3), offender-victim (T4), offender-bystander (T5), victim-bystander (T6), offender-victim-bystander (T7), non-involved (T8)].

H_1 : There are significant differences in the mean Onespread of traditional school bullying across the eight types of involvement in cyber bullying.

```
R Console
> ## oneway ANOVA(Type * bullying)
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="type_anova.txt",header=T)
> attach(cyber_bullying)
> tapply(bullying, Type, mean)
   T1      T2      T3      T4      T5      T6      T7      T8
0.8248511 0.8757067 0.7669860 0.9225933 0.8545994 0.9159899 0.8674699 0.5562862
> tapply(bullying, Type, sd)
   T1      T2      T3      T4      T5      T6      T7      T8
0.3855580 0.3579392 0.4417147 0.2883984 0.3803527 0.3396073 0.3802770 0.5079543
> sel=aov(bullying~Type,data=cyber_bullying)
> summary(sel)
   Df Sum Sq Mean Sq F value Pr(>F)
Type       7 20170  2881.5 12572 <2e-16 ***
Residuals 1210558 277448     0.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> bartlett.test(bullying~Type,data=cyber_bullying)

Bartlett test of homogeneity of variances

data:  bullying by Type
Bartlett's K-squared = 42230, df = 7, p-value < 2.2e-16
> |
```

Interpretation: Among the eight types of involvement with cyber bullying, Type 4 (offender-victim) had the highest mean of traditional school bullying (0.92). ANOVA results indicated that there were significant differences in the means of traditional school bullying across the eight types of involvement ($F = 12,572, p <.001$). The results of the Bartlett test for equal-variance ($B = 42,230, p <.001$) rejected the null hypothesis, indicating that there were unequal-variances across the eight types of involvement.

```
R Console
> tukey=glht(sel,linfct = mcp(Type='Tukey'))
> summary(tukey)
Warning in RET$pf(function("adjusted", ...):
  Completion with error > abseps

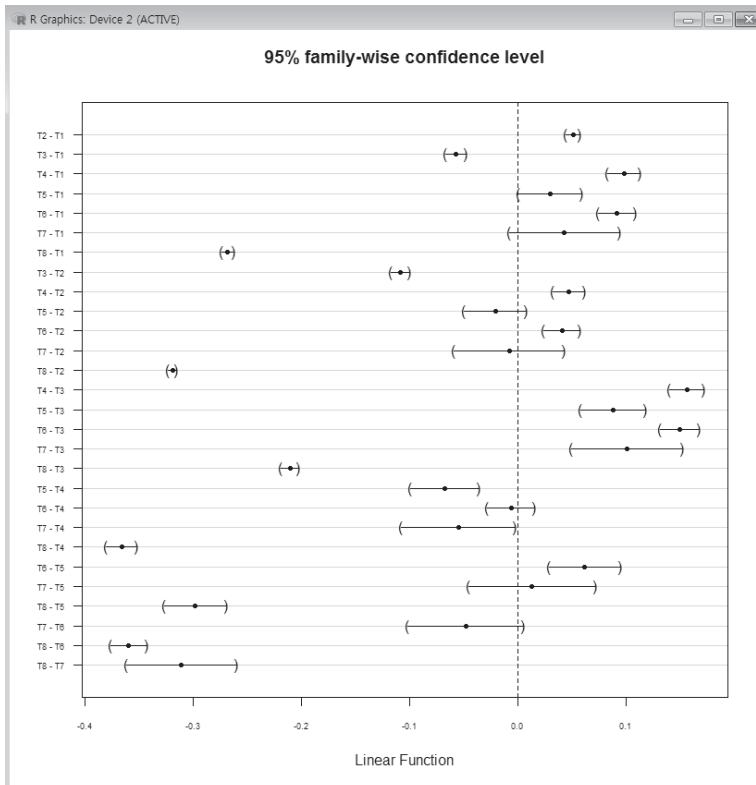
Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = bullying ~ Type, data = cyber_bullying)

Linear Hypotheses:
Estimate Std. Error t value Pr(>|t|)
T2 - T1 == 0 0.050856 0.002183 23.294 <0.001 ***
T3 - T1 == 0 -0.057865 0.003266 -17.715 <0.001 ***
T4 - T1 == 0 0.097742 0.005203 18.786 <0.001 ***
T5 - T1 == 0 0.029748 0.010030 2.966 0.0407 *
T6 - T1 == 0 0.091139 0.005961 15.288 <0.001 ***
T7 - T1 == 0 0.042619 0.017614 2.420 0.1736
T8 - T1 == 0 -0.268565 0.001923 -139.689 <0.001 ***
T3 - T2 == 0 -0.108721 0.002923 -37.199 <0.001 ***
T4 - T2 == 0 0.046887 0.004994 9.388 <0.001 ***
T5 - T2 == 0 -0.021107 0.009923 -2.127 0.3175
T6 - T2 == 0 0.040283 0.005780 6.969 <0.001 ***
T7 - T2 == 0 -0.008237 0.017554 -0.469 0.9996
T8 - T2 == 0 -0.319421 0.001253 -255.008 <0.001 ***
T4 - T3 == 0 0.155607 0.005554 28.018 <0.001 ***
T5 - T3 == 0 0.087613 0.010217 8.576 <0.001 ***
T6 - T3 == 0 0.149004 0.006270 23.764 <0.001 ***
T7 - T3 == 0 0.100484 0.017721 5.670 <0.001 ***
T8 - T3 == 0 -0.210700 0.002734 -77.080 <0.001 ***
T5 - T4 == 0 -0.067994 0.010990 -6.187 <0.001 ***
T6 - T4 == 0 -0.006603 0.007464 -0.885 0.9804
T7 - T4 == 0 -0.055123 0.018178 -3.032 0.0338 *
T8 - T4 == 0 -0.366307 0.004886 -74.972 <0.001 ***
T6 - T5 == 0 0.061391 0.011369 5.400 <0.001 ***
T7 - T5 == 0 0.012870 0.020099 0.640 0.9972
T8 - T5 == 0 -0.298313 0.009869 -30.226 <0.001 ***
T7 - T6 == 0 -0.048520 0.018409 -2.636 0.1026
T8 - T6 == 0 -0.359704 0.005687 -63.252 <0.001 ***
T8 - T7 == 0 -0.311184 0.017523 -17.758 <0.001 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)

> plot(tukey, cex.axis=0.6)
> |
```

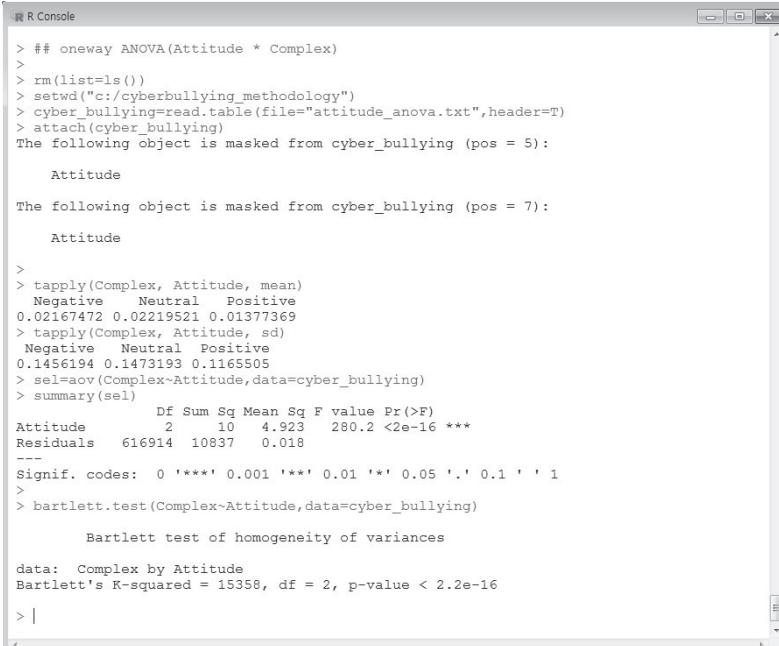


Interpretation: The results of the Tukey multiple comparisons grouped the types into [offender, offender-bystander, offender-victim-bystander] and [offender-victim, victim-bystander] ($p > .1$), indicating that there were differences in the means across the various types ($p < .001$). Thus, it is possible to group the types of involvement in cyber bullying into the following five categories: [offender, victim (victim, offender-bystander, offender-victim-bystander), bystander, complex (offender-victim, victim-bystander), and non-involved].

Research Hypothesis: ($H_0: \mu_1 - \mu_2 - \dots - \mu_k = 0$, $H_1: \mu_1 - \mu_2 - \dots - \mu_k \neq 0$)

H_0 : There are no significant differences in the means of the Onespread of the Complex-type across various attitudes to cyber bullying (Negative, Neutral, Positive).

H_1 : There are significant differences in the means of the Onespread of the Complex-type across various attitudes to cyber bullying.



```

R Console

> ## oneway ANOVA(Attitude * Complex)
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="attitude_anova.txt",header=T)
> attach(cyber_bullying)
The following object is masked from cyber_bullying (pos = 5):
      Attitude

The following object is masked from cyber_bullying (pos = 7):
      Attitude

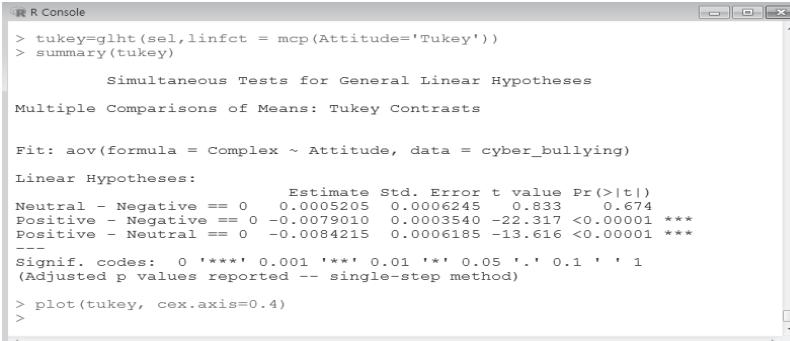
>
> tapply(Complex, Attitude, mean)
  Negative    Neutral   Positive 
0.02167472  0.02219521  0.01377369
> tapply(Complex, Attitude, sd)
  Negative    Neutral   Positive 
0.1456194  0.1473193  0.1165505
> sel=aov(Complex~Attitude,data=cyber_bullying)
> summary(sel)
   Df Sum Sq Mean Sq F value Pr(>F)
Attitude       2     10   4.923   280.2 <2e-16 ***
Residuals  616914 10837   0.018
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> bartlett.test(Complex~Attitude,data=cyber_bullying)

Bartlett test of homogeneity of variances

data: Complex by Attitude
Bartlett's K-squared = 15358, df = 2, p-value < 2.2e-16
> |

```

Interpretation: Of the attitudes to cyber bullying, ‘Neutral’ had the highest mean (0.022) among the Complex-type of involvement. ANOVA results indicated differences in the means of the Complex-type across the various types of attitudes ($F = 280.2, p <.001$). The results of the Bartlett test for equal-variance ($B = 15,358, p <.001$) rejected the null hypothesis, indicating that there were unequal-variances across the different types of attitudes to cyber bullying.



```

R Console

> tukey=glht(sel,linfct = mcp(Attitude='Tukey'))
> summary(tukey)

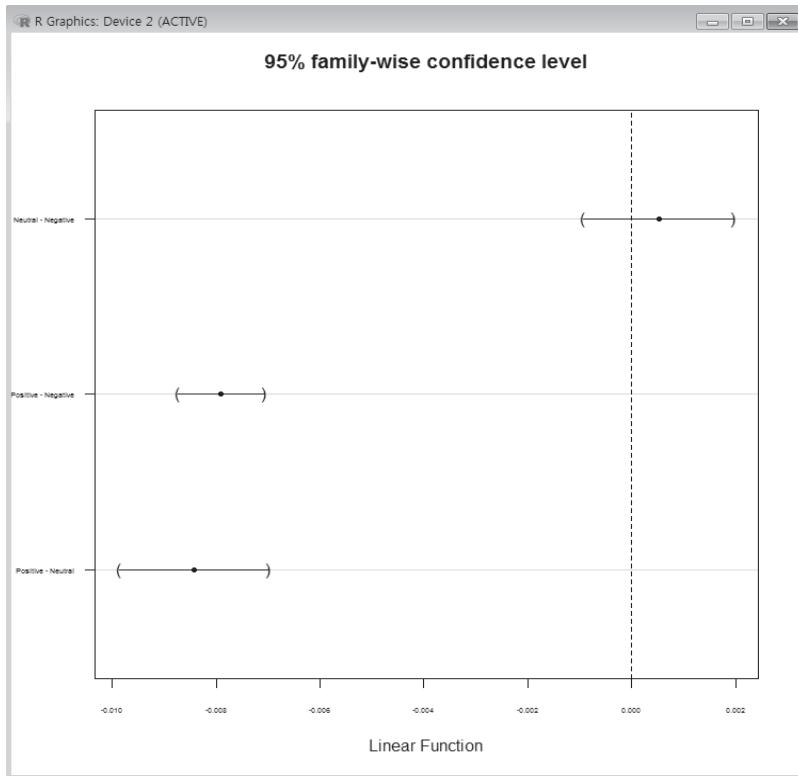
Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = Complex ~ Attitude, data = cyber_bullying)

Linear Hypotheses:
Estimate Std. Error t value Pr(>|t|)
Neutral - Negative == 0 0.0005205 0.0006245 0.833 0.674
Positive - Negative == 0 -0.0079010 0.0003540 -22.317 <0.00001 ***
Positive - Neutral == 0 -0.0084215 0.0006185 -13.616 <0.00001 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
> plot(tukey, cex.axis=0.4)
>

```



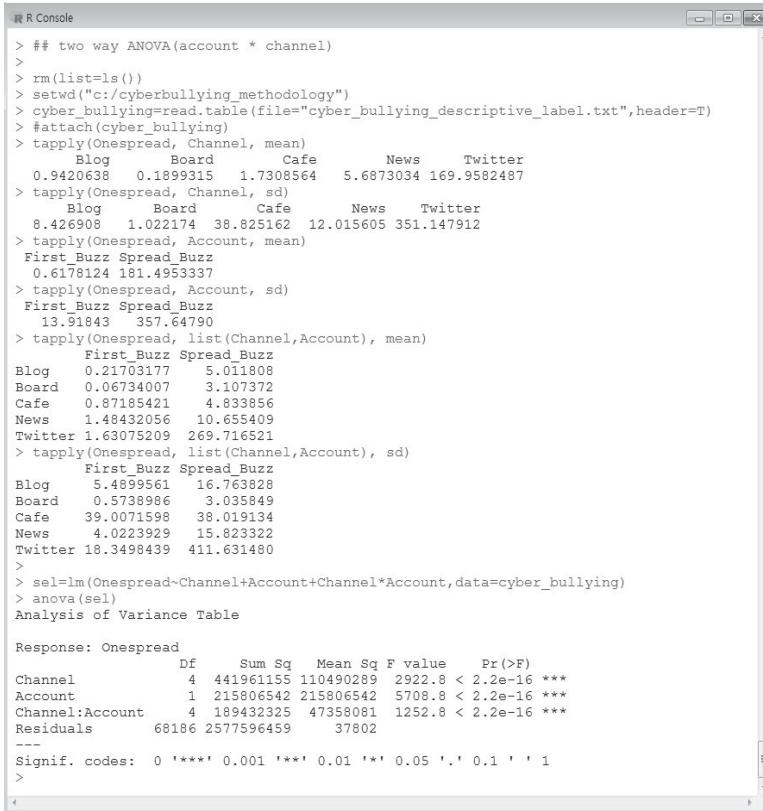
Interpretation: The results of the Dunnett multiple comparisons grouped (Negative, Neutral) as the same attitudes ($p > .1$), and indicated that there was a difference between the Negative and Positive groups ($p < .001$). Thus, the attitudes on cyber bullying may be grouped into the following two categories: [Negative (Negative, Neutral), Positive].

⑨ Test of Means (Two-Way ANOVA)

Two-way ANOVA is a method of comparing the means of two dependent variables (factors). First, a test is conducted to identify whether there is an interaction effect between the two factors. If not, it is possible to analyze the effects of each factor separately.

Research Question: Are there differences in the level of the dependent variable (Onespread) depending on Account (First, Spread) and Channel (Board, Blog, Cafe, News, Twitter)? Is there an interaction effect between Account and Channel?

```
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)
> attach(cyber_bullying)
> tapply(Onespread, Channel, mean)
  - Compute the means of Onespread by Channel.
> tapply(Onespread, Channel, sd)
> tapply(Onespread, Account, mean)
> tapply(Onespread, Account, sd)
> tapply(Onespread, list(Channel,Account), mean)
> tapply(Onespread, list(Channel,Account), sd)
  - Compute the standard deviation of Onespread by Channel and
    Account.
> sel=lm(Onespread~Channel+Account+Channel*Account,data=cyber_bullying)
  - To test for between-subjects effect, run a regression.
> anova(sel): Test for between-subjects effect.
```



```

> ## two way ANOVA(account * channel)
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_label.txt",header=T)
> #attach(cyber_bullying)
> tapply(Onespread, Channel, mean)
   Blog    Board    Cafe      News    Twitter
 0.9420638  0.1899315  1.7308564  5.6873034 169.9582487
> tapply(Onespread, Channel, sd)
   Blog    Board    Cafe      News    Twitter
 8.426908  1.022174  38.825162 12.015605 351.147912
> tapply(Onespread, Account, mean)
First_Buzz Spread_Buzz
0.6178124 181.495337
> tapply(Onespread, Account, sd)
First_Buzz Spread_Buzz
13.91843 357.64790
> tapply(Onespread, list(Channel,Account), mean)
First_Buzz Spread_Buzz
Blog 0.21703177 5.011808
Board 0.06734007 3.107372
Cafe 0.87185421 4.833856
News 1.48432056 10.655409
Twitter 1.63075209 269.716521
> tapply(Onespread, list(Channel,Account), sd)
First_Buzz Spread_Buzz
Blog 5.4899561 16.763828
Board 0.5738986 3.035849
Cafe 39.0071598 38.019134
News 4.0223929 15.823322
Twitter 18.3498439 411.631480
>
> sel=lm(Onespread~Channel+Account+Channel*Account,data=cyber_bullying)
> anova(sel)
Analysis of Variance Table

Response: Onespread
            Df    Sum Sq   Mean Sq F value    Pr(>F)
Channel        4 441961155 110490289 2922.8 < 2.2e-16 ***
Account        1 215806542 215806542 5708.8 < 2.2e-16 ***
Channel:Account 4 189432325 47358081 1252.8 < 2.2e-16 ***
Residuals     68186 2577596459   37802
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

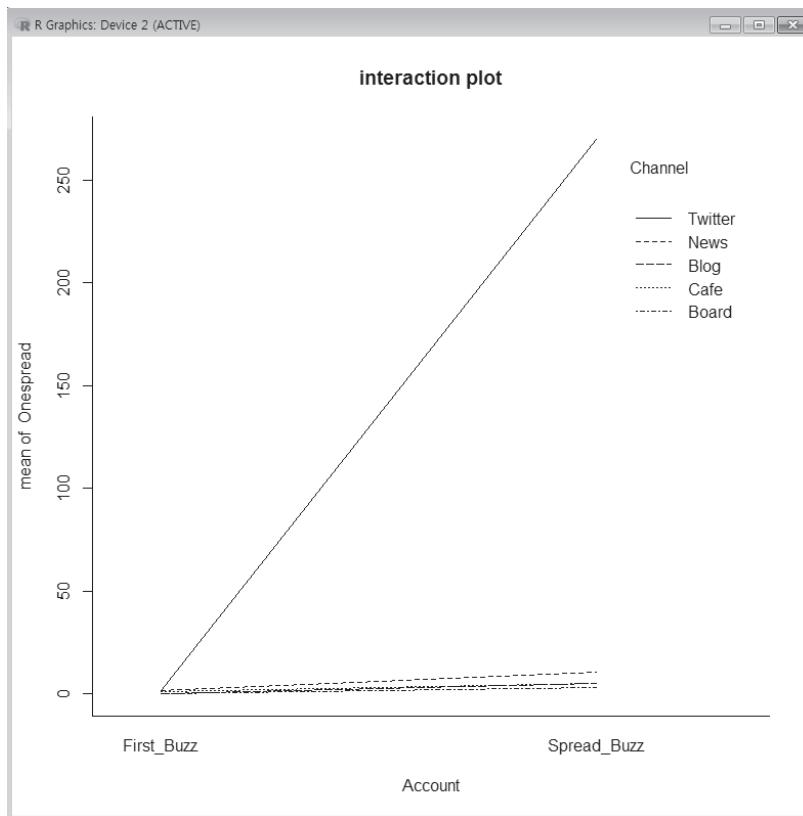
Interpretation: The analysis of the effects of Account and Channel on the one-week spread (Onespread) indicated a significant difference for both Channel ($F = 2922.8, p < .001$) and Account ($F = 5708.8, p < .001$), and an interaction effect was found between Account and Channel ($F = 1252.8, p < .001$). For all channels, the mean Onespread was higher in the case of the first postings (First) compared to the spread postings (Spread).

■ Interaction plot

```

> interaction.plot(Account, Channel, Onespread, bty='l', main='interaction
plot')
  - Create a profile diagram of Channel and Account.
  - bty(box plot type) specifies the shape of the box that surrounds the
  plot area. Here, we use (c, n, o, 7, u, l).

```



Interpretation: In the interaction plot, Twitter is crossed with other Channels, indicating the presence of an interaction effect, while (Board, Blog, Cafe, News) have no crossings among them, indicating that they belong to the same group.

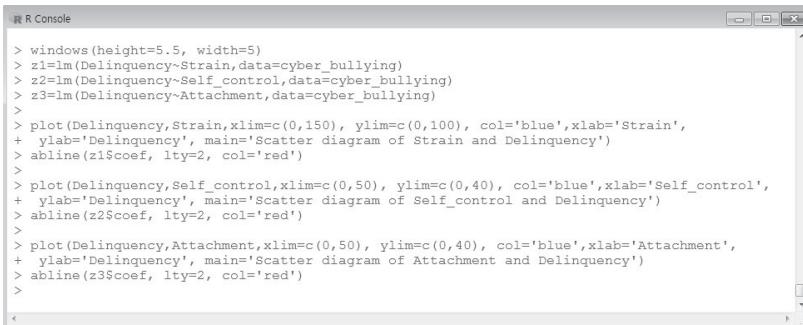
⑩ Scatter Diagram

This is conducted initially when examining the linear relationship between two continuous variables. After plotting the scatter diagram of the data of the two variables, we conduct a simple regression analysis that indicates their linear relationship.

```

> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)
> attach(cyber_bullying)
> windows(height=5.5, width=5)
    - Set the size of the output window.
> z1=lm(Delinquency~Strain,data=cyber_bullying)
    - Regress Delinquency on Strain, and assign the results to object z1.
> z2=lm(Delinquency~Self_control,data=cyber_bullying)
> z3=lm(Delinquency~Attachment,data=cyber_bullying)
> plot(Delinquency,Strain,xlim=c(0,150), ylim=c(0,100), col='blue',xlab=
    'Strain',ylab='Delinquency', main='Scatter diagram of Strain and
    Delinquency')
    - Plot a scatter diagram of Delinquency and Strain.
> abline(z1$coef, lty=2, col='red')
    - Plot the line according to the regression coefficients in z1.
    - abline() can be used to add vertical, horizontal or regression lines to a
      graph.

```

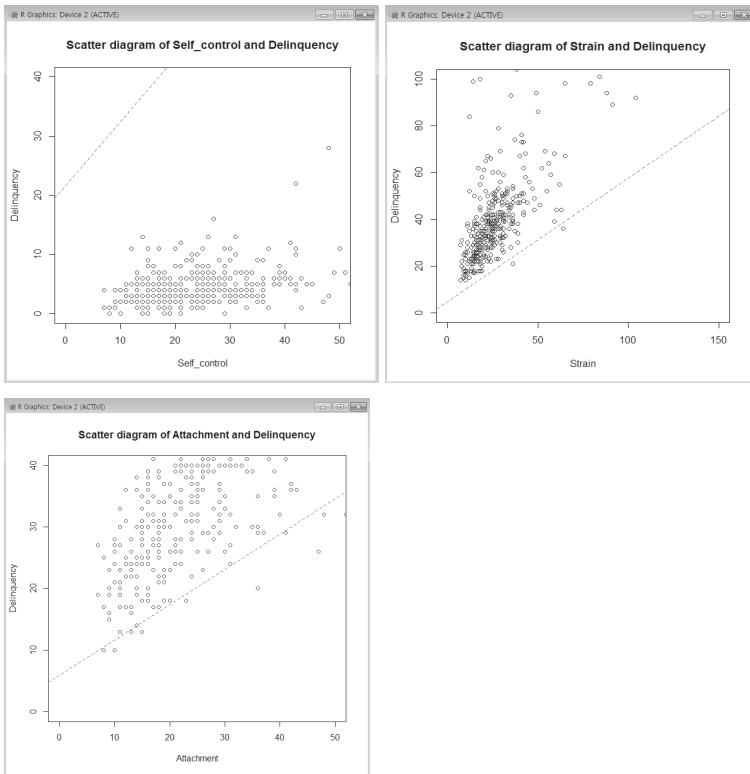


The screenshot shows the R Console window with the following code:

```

> windows(height=5.5, width=5)
> z1=lm(Delinquency~Strain,data=cyber_bullying)
> z2=lm(Delinquency~Self_control,data=cyber_bullying)
> z3=lm(Delinquency~Attachment,data=cyber_bullying)
>
> plot(Delinquency,Strain,xlim=c(0,150), ylim=c(0,100), col='blue',xlab='Strain',
+       ylab='Delinquency', main='Scatter diagram of Strain and Delinquency')
> abline(z1$coef, lty=2, col='red')
>
> plot(Delinquency,Self_control,xlim=c(0,50), ylim=c(0,40), col='blue',xlab='Self_control',
+       ylab='Delinquency', main='Scatter diagram of Self_control and Delinquency')
> abline(z2$coef, lty=2, col='red')
>
> plot(Delinquency,Attachment,xlim=c(0,50), ylim=c(0,40), col='blue',xlab='Attachment',
+       ylab='Delinquency', main='Scatter diagram of Attachment and Delinquency')
> abline(z3$coef, lty=2, col='red')
>

```



Interpretation: The scatter diagram for (Strain, Self_control, Attachment) and Delinquency indicate a positive (+) linear relationship, showing that increases in Strain, Self_control, and Attachment are associated with increases in Delinquency.

⑪ Correlation Analysis

Correlation analysis is a method for identifying whether a linear relationship exists between two quantitative variables, and provides a measure of the strength of the correlation. Using this, it is possible to measure how closely related the two variables are.

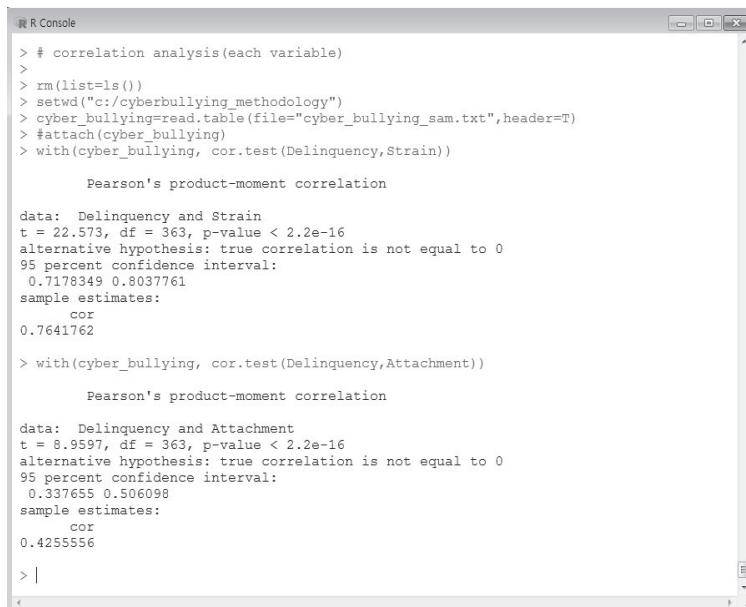
The correlation coefficient may take a value from -1 to 1, with its magnitude indicating the strength of the association. If the absolute value of the correlation coefficient is large, this indicates a close relationship between the two variables. A positive value (+) indicates a positive

relationship, a negative value (-) indicates a negative relationship, while a '0' value indicates that the two variables are not correlated. Correlation is not indicative of causality—it is a measure of the degree of association between variables.

Depending on the sample size, correlation analysis may be conducted via parametric or nonparametric methods. In general, parametric methods are used for sample sizes larger than 30. The Pearson correlation coefficient is used for parametric analysis, while the Spearman correlation measure or Kendall's tau are used for nonparametric analyses.

■ Analysis of correlation between two variables

```
> rm(list=ls()): Initialize all variables.  
> setwd("c:/cyberbullying_methodology"): Set the working directory.  
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)  
  - Assign the data file to cyber_bullying.  
> attach(cyber_bullying): Attach 'cyber_bullying' as execution data.  
> with(cyber_bullying, cor.test(Delinquency,Strain))  
  - Compute the correlation coefficient for Delinquency and Strain, and  
    compute its p-value.  
> with(cyber_bullying, cor.test(Delinquency,Attachment))
```



The screenshot shows the R Console window with the following text:

```
R Console  
> # correlation analysis(each variable)  
>  
> rm(list=ls())  
> setwd("c:/cyberbullying_methodology")  
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)  
> #attach(cyber_bullying)  
> with(cyber_bullying, cor.test(Delinquency,Strain))  
  
Pearson's product-moment correlation  
  
data: Delinquency and Strain  
t = 22.573, df = 363, p-value < 2.2e-16  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
 0.7178349 0.8037761  
sample estimates:  
       cor  
0.7641762  
  
> with(cyber_bullying, cor.test(Delinquency,Attachment))  
  
Pearson's product-moment correlation  
  
data: Delinquency and Attachment  
t = 8.9597, df = 363, p-value < 2.2e-16  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
 0.337655 0.506098  
sample estimates:  
       cor  
0.4255556  
  
> |
```

Interpretation: Delinquency and Strain were found to be strongly positively correlated (.764, $p <.001$). Delinquency and Attachment were also found to be strongly positively correlated (.425, $p <.001$).

■ Analysis of the correlation between all variables

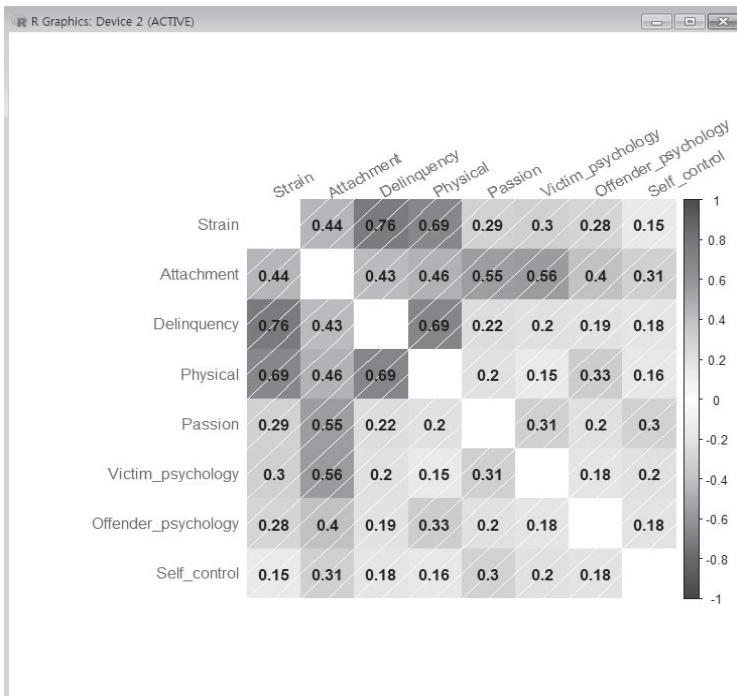
```
> cyber_bullying1=cbind(Strain,Physical,Victim_psychology,Self_control,
  Attachment,Passion,Offender_psychology,Delinquency)
  - Save the data frame Strain~Delinquency to cyber_bullying1.
> install.packages("psych")
  - Install the psychometric package, "psych".
> library(psych)
> corr.test(cyber_bullying1): Run correlation analysis for all variables.
```

```
R Console
> # correlation analysis(all variables)
> cyber_bullying1=cbind(Strain,Physical,Victim_psychology,Self_control,Attachment,
+   Passion,Offender_psychology,Delinquency)
> install.packages("psych")
Warning: package "psych" is in use and will not be installed
> library(psych)
> corr.test(cyber_bullying1)
Call:corr.test(x = cyber_bullying1)
Correlation matrix
  Strain Physical Victim_psychology Self_control Attachment Passion Offender_psychology Delinquency
Strain      1.00     0.69      0.30     0.15      0.44      0.29      0.28      0.76
Physical    0.69     1.00     0.15     0.45     0.40     0.20     0.33     0.69
Victim_psychology 0.30     0.15     1.00     0.20     0.56     0.31     0.18     0.20
Self_control 0.15     0.16     0.20     1.00     0.31     0.30     0.18     0.18
Attachment  0.44     0.46     0.56     0.31     1.00     0.55     0.40     0.43
Passion     0.29     0.20     0.31     0.30     0.55     1.00     0.20     0.22
Offender_psychology 0.28     0.33     0.18     0.18     0.40     0.20     1.00     0.19
Delinquency 0.76     0.69     0.20     0.18     0.43     0.22     0.19     1.00
Sample Size
[1] 500
Probability values (Entries above the diagonal are adjusted for multiple tests.)
  Strain Physical Victim_psychology Self_control Attachment Passion Offender_psychology Delinquency
Strain      0.00     0.00      0.00     0.01      0       0       0       0
Physical    0.00     0.00      0.01     0.01      0       0       0       0
Victim_psychology 0.00     0.01      0.00     0.00      0       0       0       0
Self_control 0.00     0.00      0.00     0.00      0       0       0       0
Attachment  0.00     0.00      0.00     0.00      0       0       0       0
Passion     0.00     0.00      0.00     0.00      0       0       0       0
Offender_psychology 0.00     0.00      0.00     0.00      0       0       0       0
Delinquency 0.00     0.00      0.00     0.00      0       0       0       0
To see confidence intervals of the correlations, print with the short=FALSE option
>
>
```

Interpretation: Examine the correlation coefficient matrix (0.69~0.19) for Strain~Delinquency, and the p -values ($p <.01$).

- The corplot package can be used for creating correlation plots.

```
R Console
> ## correlation coefficient plot(corrplot)
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)
> #attach(cyber_bullying)
> cyber_bullying1=cbind(Strain,Physical,Victim_psychology,Self_control,Attachment,
+ Passion,Offender_psychology,Delinquency)
> cyber_bullying_corr=cor(cyber_bullying1, use='pairwise.complete.obs')
> install.packages('corrplot')
Warning: package 'corrplot' is in use and will not be installed
> library(corrplot)
> corrplot(cyber_bullying_corr,
+           method="shade",
+           addshade="all",
+           tl.col="red",
+           tl.srt=30,
+           diag=FALSE,
+           addCoef.col="black",
+           order="FPC"    # "FPC": First Principle Component
+ )
>
> |
```

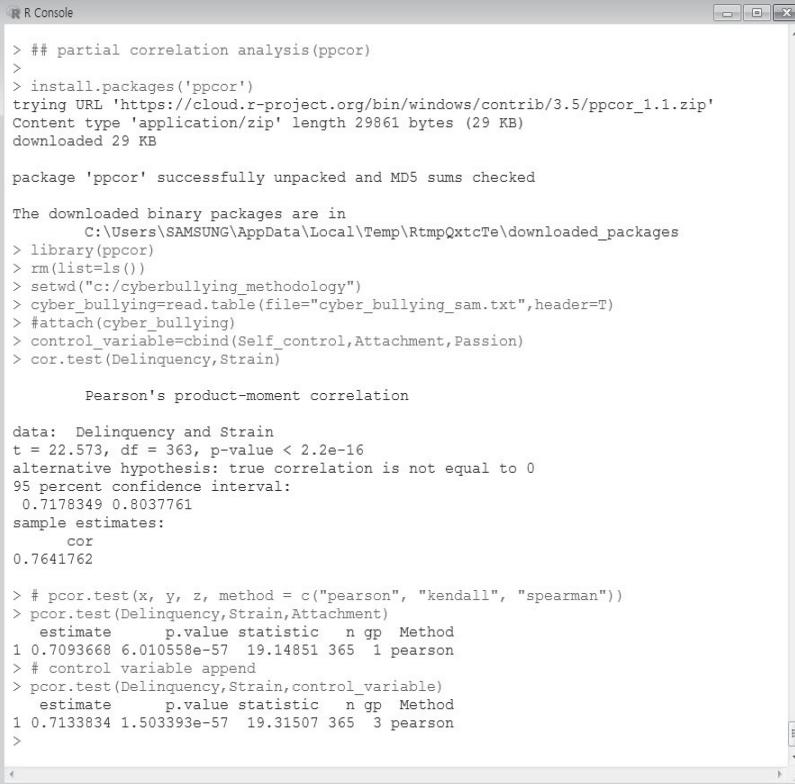


Interpretation: All factors were found to be strongly positively correlated among themselves, with the strongest correlation being observed between Delinquency and Strain.

⑫ Partial Correlation Analysis

Partial correlation analysis is similar to simple correlation analysis in that it examines the correlation between two variables. The difference is that it controls for certain variables that influence the two variables during the analysis. For instance, computing the Pearson correlation coefficient for Delinquency and Strain returns a high value, as they are influenced by Attachment. Thus, a partial correlation analysis may be used to control for Attachment, in order to examine the pure degree of association between Delinquency and Strain.

```
> install.packages('ppcor')
  - Install the partial correlation analysis package (ppcor).
> library(ppcor)
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)
> attach(cyber_bullying)
> control_variable=cbind(Self_control,Attachment,Passion)
  - Assign (Self_control, Attachment, Passion) as the control variables.
> cor.test(Delinquency,Strain)
  - Conduct correlation analysis for Delinquency and Strain.
> pcor.test(Delinquency,Strain,Attachment)
  - pcor.test(x, y, z, method = c("pearson", "kendall", "spearman"))).
  - Conduct partial correlation analysis by controlling for Attachment.
> pcor.test(Delinquency,Strain,control_variable)
  - Conduct partial correlation analysis by controlling for the three
    variables (Self_control, Attachment, Passion).
```



```
R Console

> ## partial correlation analysis(ppcor)
>
> install.packages('ppcor')
trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.5/ppcor_1.1.zip'
Content type 'application/zip' length 29861 bytes (29 KB)
downloaded 29 KB

package 'ppcor' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:/Users/SAMSUNG/AppData/Local/Temp/RtmpQxtcTe/downloaded_packages
> library(ppcor)
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying.sam.txt",header=T)
> #attach(cyber_bullying)
> control_variable=cbind(Self_control,Attachment,Passion)
> cor.test(Delinquency,Strain)

Pearson's product-moment correlation

data: Delinquency and Strain
t = 22.573, df = 363, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.7178349 0.8037761
sample estimates:
cor
0.7641762

> # pcor.test(x, y, z, method = c("pearson", "kendall", "spearman"))
> pcor.test(Delinquency,Strain,Attachment)
  estimate   p.value statistic n gp Method
1 0.7093668 6.010558e-57 19.14851 365  1 pearson
> # control variable append
> pcor.test(Delinquency,Strain,control_variable)
  estimate   p.value statistic n gp Method
1 0.7133834 1.503393e-57 19.31507 365  3 pearson
>
```

Interpretation: With Attachment controlled for, the partial correlation coefficient between Delinquency and Strain is 0.709 ($p < .001$) and is smaller than the value of the Pearson correlation coefficient, 0.764 ($p < .001$), which is a simple (i.e., uncontrolled) correlation measure. Thus, the total effect between Delinquency and Strain is .764, and the effect due to Attachment (indirect effect) is .055, while the direct effect between Delinquency and Strain is 0.709. With the three variables (Self_control, Attachment, Passion) controlled for, the partial correlation coefficient between Delinquency and Strain is 0.713 ($p < .001$), with the effect due to the three control variables found to be .051.

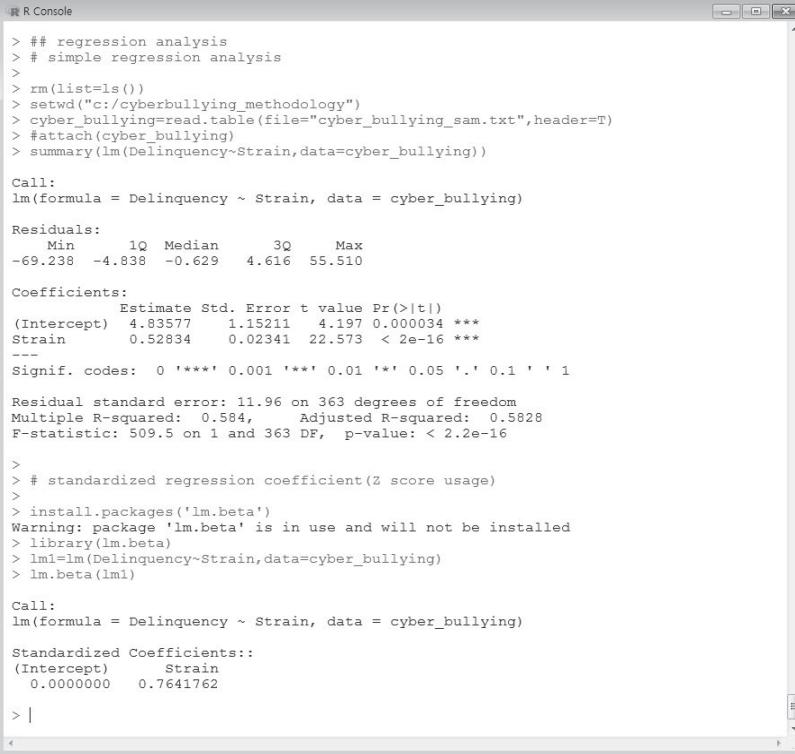
⑬ Simple Regression Analysis

Regression is a concept that expands on correlation analysis and ANOVA. It is a statistical analysis method that specifies the relationship between two observed continuous variables in the form of a mathematical equation ($Y = aX + b$). Regression analysis involves examining the relationship between the dependent and independent variables in terms of functions, with several variations depending on the number of independent variables and the scale of the dependent variable, as follows.

- Simple regression analysis: One continuous independent variable, one continuous dependent variable.
- Multiple regression analysis: Two or more continuous independent variables, one continuous dependent variable.
- Binary logistic regression analysis: One or more continuous independent variables, one binary dependent variable.
- Multinomial logistic regression analysis: One or more continuous independent variables, one multinomial dependent variable.

Research Question: In cyber bullying, does Strain have an effect on Delinquency?

```
> rm(list=ls()): Initialize all variables.  
> setwd("c:/cyberbullying_methodology"): Set the working directory.  
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)  
> attach(cyber_bullying)  
> summary(lm(Delinquency~Strain,data=cyber_bullying))  
    - Run a simple regression analysis.  
    - lm(): the function used for regression analysis.  
> install.packages('lm.beta')  
    - Install the "lm.beta" package for computing standardized regression  
      coefficients.  
> library(lm.beta)  
> lm1=lm(Delinquency~Strain,data=cyber_bullying)  
    - Run a simple regression and assign the output to the lm1 object.  
> lm.beta(lm1)  
    - Compute the standardized regression coefficients of the lm1 object  
      and display them on the screen.
```



The screenshot shows the R Console window with the following output:

```
> ## regression analysis
> # simple regression analysis
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying.sam.txt",header=T)
> #attach(cyber_bullying)
> summary(lm(Delinquency~Strain,data=cyber_bullying))

Call:
lm(formula = Delinquency ~ Strain, data = cyber_bullying)

Residuals:
    Min      1Q  Median      3Q     Max 
-69.238 -4.838 -0.629  4.616 55.510 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.83577   1.15211  4.197 0.000034 ***
Strain       0.52834   0.02341 22.573 < 2e-16 *** 
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.96 on 363 degrees of freedom
Multiple R-squared:  0.584,  Adjusted R-squared:  0.5828 
F-statistic: 509.5 on 1 and 363 DF,  p-value: < 2.2e-16

> # standardized regression coefficient(Z score usage)
>
> install.packages('lm.beta')
Warning: package 'lm.beta' is in use and will not be installed
> library(lm.beta)
> lm1=lm(Delinquency~Strain,data=cyber_bullying)
> lm.beta(lm1)

Call:
lm(formula = Delinquency ~ Strain, data = cyber_bullying)

Standardized Coefficients:
(Intercept)      Strain  
0.0000000  0.7641762 

> |
```

Interpretation: The coefficient of determination, R^2 , indicates the share of the total variation that is explained by the regression line; this indicates that 58.4% of the variation in the Delinquency factor of cyber bullying is explained by Strain. The value of R^2 may range from 0 to 1, with values closer to 1 indicating that the regression line fits the sample data more closely. The F statistic tests whether the regression equation is significant. Here, the F statistic is 509.5 with a p-value of $p = .000 < .001$, indicating a high degree of significance. The estimated equation is $\text{Delinquency} = 4.836 + 0.528\text{Strain}$, and both the intercept term and the regression coefficient are very statistically significant ($p < .001$). Standardized regression coefficients report the standardized values of all the coefficients included in the regression analysis, such that their magnitudes may be compared. A larger standardized regression coefficient indicates a greater degree of influence on the dependent variable. The standardized regression line for this example is $\text{Delinquency} = 0.764\text{Strain}$. This implies that a unit increase in Strain is associated with a .764 increase in Delinquency.

```
> anova(lm(Delinquency~Strain,data=cyber_bullying))
- Compute the ANOVA table of the regression equation.
```

```
R Console
> # ANOVA table
>
> anova(lm(Delinquency~Strain,data=cyber_bullying))
Analysis of Variance Table

Response: Delinquency
          Df Sum Sq Mean Sq F value    Pr(>F)
Strain      1  72831   72831 509.52 < 2.2e-16 ***
Residuals 363  51887    143
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> |
```

Interpretation: The ANOVA table is used to examine the goodness-of-fit of the regression model. In the case of a simple regression model, it is equivalent to the F statistic (509.5, $p < .001$). Thus, the regression equation is found to be highly significant.

■ Obtain estimates of Delinquency depending on Strain

```
> simple_reg=lm(Delinquency~Strain,data=cyber_bullying)
> Strain_new=seq(50, 500, 50)
- Assign a sequence of values, from 50 to 500 in increments of 50, to the Strain_new object.
> Delinquency_new=predict(simple_reg, newdata=data.frame(Strain=Strain_new))
- Compute Delinquency estimates corresponding to the new Strain values, and assign these to the Delinquency_new object.
> Delinquency_new: Display the Delinquency estimates on the screen.
- When Strain = 50, the Delinquency estimate is 31.25.
- When Strain = 500, the Delinquency estimate is 269.00.
```

```
R Console
> # estimated value(Strain * Delinquency)
>
> simple_reg=lm(Delinquency~Strain,data=cyber_bullying)
> Strain_new=seq(50, 500, 50)
> Delinquency_new=predict(simple_reg, newdata=data.frame(Strain=Strain_new))
> Delinquency_new
     1      2      3      4      5      6      7      8 
31.25255 57.66934 84.08612 110.50290 136.91968 163.33646 189.75324 216.17002 
9      10    
242.58681 269.00359
> |
```

⑭ Multiple Regression Analysis

Multiple regression analysis is a method for examining the effects of two or more independent variables on a dependent variable. The following aspects must be considered when conducting multiple regression analysis.

- Look for dependences between the independent variables; i.e., multicollinearity. Variables with high multicollinearity (low tolerance) must be discarded. Multicollinearity refers to a phenomenon in regression analysis where the inclusion of independent variables that are highly correlated to each other causes the determinant of the variance-covariance matrix to approach 0; thus, leading to extremely poor precision in the estimated coefficients.
- The VIF (Variance Inflation Factor) is used in OLS (Ordinary Least Squares) regression estimation to test for the degree of multicollinearity. In general, VIF values must be smaller than five or 10 to ensure independence among the independent variables (Montgomery & Runger, 2003: p. 461).
- There must be no autocorrelation in the residual term; i.e., the residuals must be mutually independent.
- It is necessary to check for equivariance in the dependent and independent variables, using partial regression residual plots.
- There are largely two methods for including independent variables in multiple regression analysis.
 - Enter method: All independent variables are included at the same time; the multiple regression model can be specified at once [use the lm() function].
 - Stepwise method: A method where the regression model is specified based on tests of the statistical significance of the independent variables. Independent variables with low significance are sequentially discarded, and the multiple regression model is specified using only the ‘fittest’ variables [use the step() function].

Research Question: Which independent variables among (Strain~Offender_psychology) affect the Delinquency factor in cyber bullying?

- Multiple regression analysis via the Enter (simultaneous inclusion) method

```
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)
> attach(cyber_bullying)
> cyber_bullying1=data.frame(Strain,Physical,Victim_psychology,
  Self_control, Attachment,Passion,Offender_psychology,Delinquency)
  - Assign the dependent and independent variables to the cyber_
    bullying1 object as a data frame.
> summary(lm(Delinquency~.,data=cyber_bullying1))
  - Run a first-round multiple regression for all independent variables.
```

The screenshot shows the R Console window with the following output:

```
R Console

> ## multiple regression analysis
> ## enter method(multiple regression)
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_sam.txt",header=T)
> #attach(cyber_bullying)
> cyber_bullying1=data.frame(Strain,Physical,Victim_psychology,Self_control,Attachment,Passion,Offender_psychology,Delinquency)
> summary(lm(Delinquency~.,data=cyber_bullying1))

Call:
lm(formula = Delinquency ~ ., data = cyber_bullying1)

Residuals:
    Min      1Q  Median      3Q     Max 
-50.691  -5.113  -0.963   3.608  59.904 

Coefficients:
            Estimate Std. Error t value    Pr(>|t|)    
(Intercept) 3.12172   1.80995   1.725    0.08544 .  
Strain       0.38869   0.03160  12.302   < 2e-16 *** 
Physical     0.34582   0.05691   6.077  0.00000000314 ***
Victim_psychology -0.12620  0.08510  -1.483    0.13895  
Self_control 0.34965   0.20302   1.722    0.08588 .  
Attachment    0.16666   0.06731   2.476    0.01375 *  
Passion       -0.08066  0.06105  -1.321    0.18724  
Offender_psychology -0.22758  0.08612  -2.643    0.00859 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

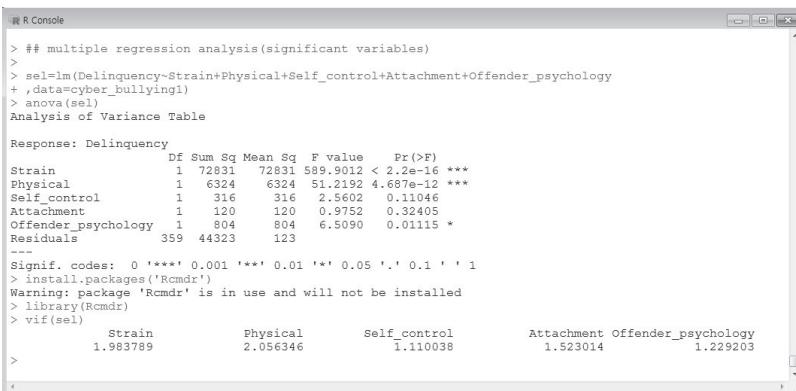
Residual standard error: 11.08 on 357 degrees of freedom
Multiple R-squared:  0.6483,  Adjusted R-squared:  0.6414 
F-statistic: 94.01 on 7 and 357 DF,  p-value: < 2.2e-16

>
```

Interpretation: The intercept ($B = 3.12, p <.1$), Strain ($B = 0.389, p <.001$), Physical ($B = 0.346, p <.001$), Self_control ($B = 0.349, p <.1$), and Attachment ($B = 0.167, p <.05$) were found to have a positive effect on Delinquency. In contrast, Offender_psychology ($B = -0.228, p <.01$) was found to have a negative effect on Delinquency. Victim_psychology and Passion were found to have no effect on Delinquency. The significance of the regression equation, as measured by the F-value (94.01, $p <.001$), indicated that the regression equation was very highly significant.

★ Second-round multiple regression analysis

```
> sel=lm(Delinquency~Strain+Physical+Self_control+Attachment+
  Offender_psychology, data=cyber_bullying1)
  - Run a multiple regression using only the significant independent
    variables.
> anova(sel): Test the regression coefficients (ANOVA by factor).
> install.packages('Rcmdr')
  - Install the "Rcmdr" package, which includes the VIF function.
> library(Rcmdr)
> vif(sel)
  - Run a diagnostic (VIF) for multicollinearity among the independent
    variables.
```



The screenshot shows the R Console window with the following output:

```
R Console
> ## multiple regression analysis(significant variables)
>
> sel=lm(Delinquency~Strain+Physical+Self_control+Attachment+Offender_psychology
+ , data=cyber_bullying1)
> anova(sel)
Analysis of Variance Table

Response: Delinquency
            Df Sum Sq Mean Sq F value    Pr(>F)
Strain          1 72831  72831 589.9012 < 2.2e-16 ***
Physical        1   6324   6324  51.2192 4.687e-12 ***
Self_control    1    316    316  2.5602  0.11046
Attachment       1    120    120  0.9752  0.32405
Offender_psychology 1    804    804  6.5090  0.01115 *
Residuals      359 44323   123
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> install.packages('Rcmdr')
Warning: package 'Rcmdr' is in use and will not be installed
> library(Rcmdr)
> vif(sel)
              Strain           Physical          Self_control          Attachment Offender_psychology
1.983789        2.056346        1.110038        1.523014        1.229203
>
>
```

Interpretation: Run a third-round regression after discarding the variable with the higher VIF value (Attachment), from among the variables (Self_control, Attachment) from the second-round regression that were not found to be significant.

★ Third-round multiple regression analysis

```

> ## multiple regression analysis(significant variables) final
> summary(lm(Delinquency~Strain+Physical+Self_control+Offender_psychology
+ ,data=cyber_bullying1))

Call:
lm(formula = Delinquency ~ Strain + Physical + Self_control +
    Offender_psychology, data = cyber_bullying1)

Residuals:
    Min      1Q  Median      3Q     Max 
-51.092 -5.197 -1.035  4.249 58.920 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.29291   1.43011   3.002  0.00287 **  
Strain       0.38211   0.03022  12.643 < 2e-16 ***
Physical     0.39259   0.05391   7.283 2.08e-12 ***
Self_control 0.36819   0.19520   1.886  0.06007 .    
Offender_psychology -0.18395   0.08348  -2.204  0.02819 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 11.14 on 360 degrees of freedom
Multiple R-squared:  0.642,    Adjusted R-squared:  0.6381 
F-statistic: 161.4 on 4 and 360 DF,  p-value: < 2.2e-16

```

Interpretation: In the third-round regression model, the Intercept ($B = 4.29, p <.01$), Strain ($B = 0.382, p <.001$), Physical ($B = 0.393, p <.001$), and Self_control ($B = 0.368, p <.1$) were found to have a positive effect on Delinquency. In contrast, Offender_psychology ($B = -0.184, p <.05$) was found to have a negative effect on Delinquency. The estimated regression equation was $4.29 + 0.382\text{Strain} + 0.393\text{Physical} + 0.368\text{Self_control} - .184\text{Offender_psychology}$, and its explanatory power was 63.8 (Adjusted R^2). The F statistic was 161.4 with a p -value of $p = .000 <.001$, indicating that the estimated regression equation was highly significant.

★ Computing standardized regression coefficients

- > install.packages('lm.beta')
 - Install the “lm.beta” package for computing standardized regression coefficients.
- > library(lm.beta)
- > lm1=lm(Delinquency~Strain+Physical+Self_control+Offender_psychology,data=cyber_bullying1)
 - Run a multiple regression using only the significant variables, and assign the output to the lm1 object.

```
> lm.beta(lm1)
```

- Compute the standardized regression coefficients in the lm1 object, and display them on the screen.

```
R R Console
> # standardized regression coefficient(Z score usage)
>
> install.packages('lm.beta')
Warning: package 'lm.beta' is in use and will not be installed
> library(lm.beta)
> lm1=lm(Delinquency~Strain+Physical+Self_control+Offender_psychology
+ ,data=cyber_bullying1)
> lm.beta(lm1)

Call:
lm(formula = Delinquency ~ Strain + Physical + Self_control +
    offender_psychology, data = cyber_bullying1)

Standardized Coefficients:
(Intercept)          Strain          Physical          Self_control Offender_psychology
0.000000000       0.55268344      0.32360472      0.06089151      -0.07440746

> |
```

Interpretation: The standardized regression equation, using standardized coefficients for comparisons of magnitude, was $0.552\text{Strain} + 0.324\text{Physical} + 0.061\text{Self_control} - 0.074\text{Offender_psychology}$, indicating that Strain had the greatest influence among the independent variables, followed by Physical, Offender psychology, and Self control.

★ Computing the variance inflation factor (VIF)

```
R R Console
> # VIF(variance inflation factor)
>
> install.packages('Rcmdr')
Warning: package 'Rcmdr' is in use and will not be installed
> library(Rcmdr)
> sel=lm(Delinquency~Strain+Physical+Self_control+Offender_psychology,
+ ,data=cyber_bullying1)
> anova(sel)
Analysis of Variance Table

Response: Delinquency
            Df Sum Sq Mean Sq F value    Pr(>F)
Strain         1 72831   72831 587.2795 < 2.2e-16 ***
Physical        1   6324    6324 50.9916 5.165e-12 ***
Self_control    1    316     316  2.5488  0.11125
Offender_psychology 1    602     602  4.8554  0.02819 *
Residuals     360 44645    124
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> vif(sel)
             Strain          Physical          Self_control Offender_psychology
1.921674      1.985608      1.048043      1.146734

> |
```

Interpretation: In general, an independent variable must have a VIF value smaller than 5 or 10 in order to be deemed independent of the remaining variables (Montgomery & Runger, 2003: p. 461). As all of the variables' VIF values are smaller than 10, this model does not have a multicollinearity issue.

★ Computing the tolerance

```
> tol=c(1.922,1.986,1.048,1.147)
  - Assign the VIF values of the independent variables in the sel object
    to the tol vector.
> tolerance = 1/tol
  - Compute the tolerance values of the independent variables.
> tolerance
  - Display the tolerance values of the independent variables on the screen.
```

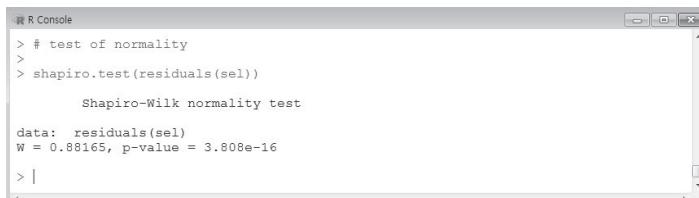


The screenshot shows the R Console window with the following text:

```
R Console
> ## tolerance function
>
> tol=c(1.922,1.986,1.048,1.147)
> tolerance = 1/tol
> tolerance
[1] 0.5202914 0.5035247 0.9541985 0.8718396
> |
```

Interpretation: Variables with lower tolerance have relatively higher multicollinearity. In the case of the regression equation here, Physical is the variable with the highest multicollinearity.

★ Testing for the normality of the residuals



The screenshot shows the R Console window with the following text:

```
R Console
> # test of normality
>
> shapiro.test(residuals(sel))

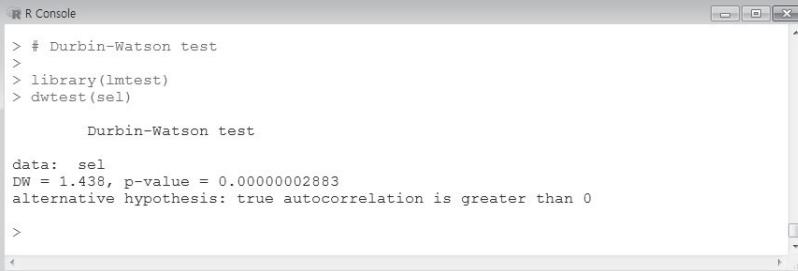
Shapiro-Wilk normality test

data: residuals(sel)
W = 0.88165, p-value = 3.808e-16
> |
```

Interpretation: Using the Shapiro-Wilk test statistic (null hypothesis: the residuals are distributed normally), the normality assumption is accepted if ' $p > \alpha$ '. Thus, under the 0.01 significance level, the regression equation estimated here (sel) rejects the null hypothesis ($p < .001$), failing to satisfy the normality assumption.

★ Testing for autocorrelation in the residuals

```
> library(lmtest)
- Load the "lmtest" package, to use the dwtest() function.
> dwtest(sel): Conduct the Durbin-Watson test.
```



The screenshot shows an R console window titled "R Console". The code entered is:

```
> # Durbin-Watson test
>
> library(lmtest)
> dwtest(sel)

Durbin-Watson test

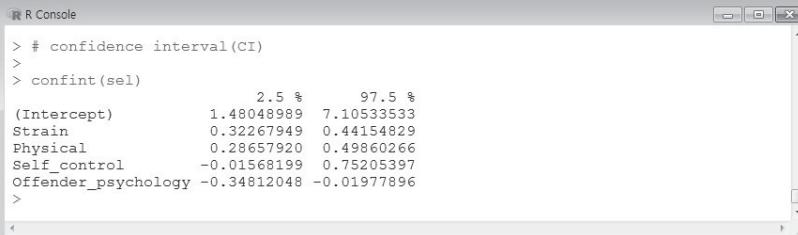
data: sel
DW = 1.438, p-value = 0.00000002883
alternative hypothesis: true autocorrelation is greater than 0

>
```

The output shows the results of the Durbin-Watson test: DW = 1.438, p-value = 0.00000002883, with the alternative hypothesis being that true autocorrelation is greater than 0.

Interpretation: The null hypothesis (the residuals of the regression model are mutually independent) is rejected ($D = 1.438$, $p <.001$), indicating the presence of autocorrelation in the residuals.

★ Compute the 95% confidence intervals for the regression coefficients.



The screenshot shows an R console window titled "R Console". The code entered is:

```
> # confidence interval (CI)
>
> confint(sel)
      2.5 %    97.5 %
(Intercept) 1.48048989 7.10533533
Strain       0.32267949 0.44154829
Physical     0.28657920 0.49860266
Self_control -0.01568199 0.75205397
Offender_psychology -0.34812048 -0.01977896
>
```

The output displays the 95% confidence intervals for each regression coefficient: (Intercept) [1.48048989, 7.10533533], Strain [0.32267949, 0.44154829], Physical [0.28657920, 0.49860266], Self_control [-0.01568199, 0.75205397], and Offender_psychology [-0.34812048, -0.01977896].

■ Model Comparisons

Models can be compared by setting up a null hypothesis (H_0 : $Y = X_1\beta_1 + \varepsilon$) and an alternative hypothesis (H_1 : $Y = X_1\beta_1 + X_2\beta_2 + \varepsilon$), and conducting F tests.

```
> fit_s=lm(Delinquency~Strain)
- Run a simple regression model with one independent variable (Strain)
  and assign the output to the fit_s ( $H_0$ ) object.
```

```
> fit_t=lm(Delinquency~Strain+Physical+Self_control+Offender_psychology,data
= cyber_bullying1)
- Run a multiple regression model and assign the output to the fit_t ( $H_1$ )
object.
> anova(fit_s, fit_t): Compare the models.
```

The screenshot shows the R Console window with the following output:

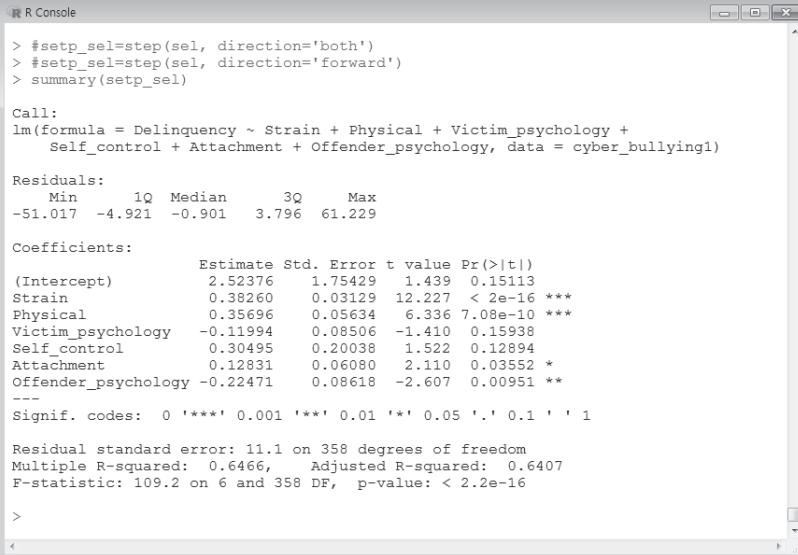
```
> ## comparison of regression models(Tip)
>
> fit_s=lm(Delinquency~Strain)
> fit_t=lm(Delinquency~Strain+Physical+Self_control+Offender_psychology,
+ data=cyber_bullying1)
> anova(fit_s, fit_t)
Analysis of Variance Table

Model 1: Delinquency ~ Strain
Model 2: Delinquency ~ Strain + Physical + Self_control + Offender_psychology
  Res.Df   RSS Df Sum of Sq    F    Pr(>F)
1     363 51887
2     360 44645  3    7241.9 19.465 1.022e-11 ***
---
Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 '.' 1
>
```

Interpretation: In this model comparison, the null hypothesis(H_0) is rejected ($F = 19.47, p <.001$) in favor of accepting fit_t(H_1) as the finalized regression model.

■ Multiple regression analysis via the Stepwise method

```
> library(MASS): Load the “MASS” package.
> sel=lm(Delinquency~.,data=cyber_bullying1)
- Run a multiple regression model with all independent variables, and
assign the output to the sel object.
> setp_sel=step(sel, direction='both')
- Run stepwise regressions on the sel object, and assign the output to
the setp_sel object.
- The ‘direction =’ option specifies the method of selecting variables
(‘both’, ‘backward’, ‘forward’).
> summary(setp_sel): Display the finalized model on the screen.
```



```
R Console

> #setp_sel=step(sel, direction='both')
> #setp_sel=step(sel, direction='forward')
> summary(setp_sel)

Call:
lm(formula = Delinquency ~ Strain + Physical + Victim_psychology +
   Self_control + Attachment + Offender_psychology, data = cyber_bullying1)

Residuals:
    Min      1Q  Median      3Q     Max 
-51.017 -4.921 -0.901  3.796  61.229 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.52376   1.75429   1.439  0.15113    
Strain       0.38260   0.03129  12.227 < 2e-16 ***
Physical     0.35696   0.05634   6.336 7.08e-10 ***
Victim_psychology -0.11994  0.08506 -1.410  0.15938    
Self_control  0.30495   0.20038   1.522  0.12894    
Attachment    0.12831   0.06080   2.110  0.03552 *  
Offender_psychology -0.22471  0.08618 -2.607  0.00951 ** 
---
Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 ' ' 1

Residual standard error: 11.1 on 358 degrees of freedom
Multiple R-squared:  0.6466,    Adjusted R-squared:  0.6407 
F-statistic: 109.2 on 6 and 358 DF,  p-value: < 2.2e-16

>
```

Interpretation: According to the regression results via the stepwise method, Strain ($B = 0.383, p <.001$), Physical ($B = 0.357, p <.001$), and Attachment ($B = 0.128, p <.05$) were found to have a positive effect on Delinquency, while Offender_psychology ($B = -0.225, p <.01$) had a negative effect on Delinquency.

15 Factor Analysis

Factor analysis is a statistical analysis method wherein the correlations among multiple variables are analyzed to group highly correlated items or covariates into several factors, thereby attributing significance to those factors. This method is used for establishing the validity of measurement instruments. Additionally, it is used in social big data analysis for reduction a large number of keywords (variables).

Validity indicates the degree to which the measurements taken via the instrument (questionnaire) faithfully measure the real concept. Thus, it indicates the accuracy with which some concept or attribute has been measured, and is examined via exploratory factor analysis or confirmatory factor analysis. The former, which is discussed in this text, is also known as traditional factor analysis. It is an exploratory method of analysis that is used for setting the research direction in the absence of a systematic or established theory. The latter, on the other hand, is used to assess the

validity of factors and variables when the causal links between them have been predetermined on the basis of very well-established theoretical backgrounds.

■ Factor Analysis Steps

- The Bartlett test is conducted to determine whether the population correlation matrix is an identity matrix (with 1 along the diagonal and all other elements equal to 0). If the null hypothesis is rejected, the correlations between the variables are deemed to be statistically significant, and thus suitable for factor analysis.
- Using the scree chart, which plots the eigenvalues obtained during the minimum factor extraction, the presence of at least one ‘broken lines’(elbow) indicates suitability for factor analysis.
- Determining the number of factors: An eigenvalue (the magnitude of the variance in the variables accounted by some factor) larger than one indicates that a factor can explain one or more variable. In general, the number of factors is determined such that their eigenvalues are at least one.
- The factor loadings indicate the correlations between the factors and the variables. In general, variables whose factor loadings have an absolute value of at least 0.4 are considered to be significant.
- Factor Rotation: This involves the rotation of the factor axis in order to clarify the variables included in the factors. Common methods include varimax and oblique rotation.

Research Question: Assess the validity of the 21 strain variables collected for the purpose of measuring the strain factors among adolescents in cyber bullying: Domestic_violence, Child_abuse, Parental_divorce, Economic_problems, Friend_Violence, Break_ups, Academic_stress, School_violence_experience, Materialism, Bullying_culture, Hell_Korea, Interested_soldier, Games, Internet_addiction, Celebrities, Movie, Adults, Gags, Chat_apps, YouTube, and Personal_broadcasting. While the strain factors of adolescents were measured to be 28 in number via the analysis in Section (Text Mining from Social Big Data Related to Cyber Bullying) of this text, keywords that appeared only in online documents during certain time periods (Female_dislike, etc.) and seven others with a low occurrence frequency (Individualism, etc.) were not included in the factor analysis.

■ Conducting the first-round factor analysis.

```
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_factor_strain28.txt",header=T)
> attach(cyber_bullying)
> fact1=cbind(Domestic_violence,Child_abuse,Parental_divorce,
  Economic_problems,Friend_Violence,Break_ups,Academic_stress,Scho
  ol_violence_experience,Materialism,Bullying_culture,Hell_Korea,Inter
  ested_soldier,Games,Internet_addiction,Celebrities,Movie,Adults,Gags,
  Chat_apps,Youtube,Personal_broadcasting)
  - Assign the 21 strain variables as a data frame to the fact1 object.
> install.packages("psych")
  - Install the "psych" package for running KMO analysis.
> library(psych)
> KMO(fact1): Run the KMO Test.
> bartlett.test(list(Domestic_violence,Child_abuse,Parental_divorce,
  Economic_problems,Friend_Violence,Break_ups,Academic_stress,
  School_violence_experience,Materialism,Bullying_culture,Hell_Korea,
  Interested_soldier,Games,Internet_addiction,Celebrities,Movie,Adults,
  Gags,Chat_apps,Youtube,Personal_broadcasting))
  - Run the Bartlett test.
```

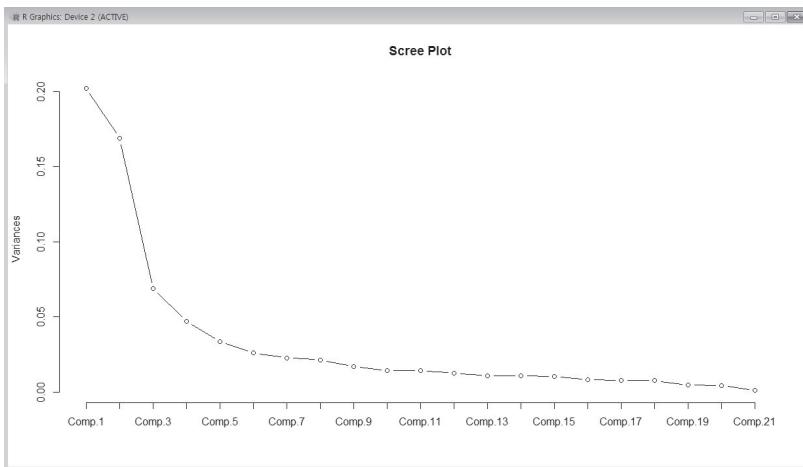
The screenshot shows the R console window with the following output:

```
> ## factor analysis (data file: cyber_bullying_factor_strain28.txt)
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_factor_strain28.txt",header=T)
> #attach(cyber_bullying)
> fact1=cbind(Domestic_violence,Child_abuse,Parental_divorce,Economic_problems,
+ Friend_Violence,Break_ups,Academic_stress,School_violence_experience,
+ Materialism,Bullying_culture,Hell_Korea,Interested_soldier,Games,Internet_addicti$+
+ Celebrities,Movie,Adults,Gags,Chat_apps,Youtube,Personal_broadcasting)
>
> ## KMO & Bartlett test
> install.packages("psych")
Warning: package 'psych' is in use and will not be installed
library(psych)
> KMO(fact1)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(x = fact1)
Overall MSA = 0.52
MSA for each item =
   Domestic_violence           Child_abuse          Parental_divorce
                           0.54                           0.53                           0.62
   Economic_problems           Friend_Violence          Break_ups
                           0.56                           0.51                           0.56
   Academic_stress            School_violence_experience       Materialism
                           0.55                           0.51                           0.53
   Bullying_culture             Hell_Korea          Interested_soldier
                           0.52                           0.57                           0.52
   Games                      Internet_addiction      Celebrities
                           0.56                           0.55                           0.75
   Movie                       Adults                     Gags
                           0.73                           0.55                           0.58
   Chat_apps                  Youtube          Personal_broadcasting
                           0.56                           0.64                           0.54
>
> |
```

Interpretation: The KMO value is 0.52, and the Bartlett test (to see whether there was zero correlation among the variables) returned significant ($p < .001$) results. Thus, we conclude that the correlation matrix is suitable for factor analysis.

★ Plotting the Scree Chart

```
> library(graphics): Load the "graphics" package.
> scr=princomp(fact1)
  - Run principle component analysis and assign the output to the scr
  object.
> screeplot(scr,npcs=21,type='lines',main='Scree Plot')
  - Plot the scree chart.
```



Interpretation: The horizontal axis of the scree chart indicates the number of factors, and the vertical axis indicates the variance of the eigenvalues. The chart's line starts flattening out to the right of factor 9 and exhibits a broken lines from factor 1 to factor 9. Thus, we found that the data is suitable for factor analysis.

★ Compute the eigenvalues of the fact1 vector (determining the number of factors).

```
> eigen(cor(fact1))$val
```

```

> # eigen value
>
> eigen(cor(fact1))$val
[1] 2.03122997 1.31896789 1.21269090 1.13724074 1.09389925 1.06094648 1.04145170 1.02892645
[9] 1.00955561 0.99690396 0.97690232 0.96345118 0.94769829 0.93333917 0.92340096 0.90178820
[17] 0.88137185 0.86061531 0.83551651 0.81400557 0.03009771
> |

```

Interpretation: The purpose of factor analysis is to reduce the number of variables. Here, nine factors were found to have eigenvalues of at least one (2.031–1.010).

★ Run the factor analysis.

```

> FA1=factanal(fact1, factors=9, rotation='none')
  - factors = 9 (As stated above, the determined number of factors whose eigenvalues, calculated using the eigen function, were greater than 1.)
  - rotation: none (no rotation), varimax (orthogonal), promax (oblique).
> FA1
> VA1=factanal(fact1, factors=9, rotation='varimax')
> VA1

```

```

> # R Console
> VA1=factanal(fact1, factors=9, rotation="varimax")
> VA1
Call:
factanal(x = fact1, factors = 9, rotation = "varimax")

Uniquenesses:
Domestic_violence          Child_abuse           Parental_divorce      Economic_problems      Friend_Violence        Break_ups
 0.782                      0.867                  0.948                 0.846                  0.005                0.975
Academic_failure             School_violence_expertise   Materialism            Bullying_culture       Hell_Korea            Interested_in_art
 0.832                      0.005                  0.891                 0.269                  0.953                0.815
Gangs                       Internets_addiction      Capitalism            Movie                 Adults               Gags
 0.848                      0.213                  0.846                 0.894                  0.980                0.986
Chat_apps                    YouTube                Personal_broadcasting
 0.918                      0.984                  0.998

Loadings:
 Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8 Factor9
Domestic_violence          0.453
Child_abuse                 0.357
Parental_divorce            0.345
Economic_problems           0.346
Friend_Violence             0.980
Hell_Korea                  0.156
Academic_failure            0.005
School_violence_expertise  0.329
Gangs                       0.101
Internets_addiction         0.368
Capitalism                  0.430
Movie                       0.490
Adults                      0.112
Break_ups                   0.170
Bullying_culture            0.174
Hell_Korea                  0.112
Materialism                 0.328
YouTube                     0.210
Personal_broadcasting       0.252
Gags                        0.101
Chat_apps                   0.056
YouTube                     0.252
Personal_broadcasting       0.114

Test of the hypothesis that 9 factors are sufficient.
The chi-square statistic is 57 degrees of freedom.
The p-value is 1.65e-73

```

Interpretation: According to the results of the first-round varimax rotated factor analysis results, variables (adolescent strain factors) with factor loadings of less than 0.25 were discarded for the second-round factor analysis. Twenty-one variables were included in the first-round analysis. Of these, eight variables (Parental_divorce, Break_ups, Bullying_culture, Hell_Korea, Adults, Gags, YouTube, Personal_broadcasting) were discarded.

★ Computing the eigenvalues for the second-round factor analysis.

```
> fact1=cbind(Domestic_violence,Child_abuse,Economic_problems,
  Friend_Violence,Academic_stress,School_violence_experience,Interested_soldier,
  Materialism,Games,Internet_addiction,Celebrities,Movie,Ch
  at_apps)
- Assign the 13 strain variables to be used in the second-round factor
  analysis as a data frame to the fact1 object.
> eigen(cor(fact1))$val
```

```
> # secondary factor analysis
>
> fact1=cbind(Domestic_violence,Child_abuse,Economic_problems,
+ Friend_Violence,Academic_stress,School_violence_experience,Interested_soldier,
+ Materialism,Games,Internet_addiction,Celebrities,Movie,Chat_apps)
>
> eigen(cor(fact1))$val
[1] 2.02306971 1.29974959 1.18004361 1.09341736 1.04288911 1.00997270 0.97975321
[8] 0.92546950 0.89032300 0.87199805 0.83677865 0.81643670 0.03009882
> |
```

Interpretation: According to the results of the second-round factor analysis conducted here, there were six factors with eigenvalues of at least 1 (2.02–1.01).

★ Second-Round Factor Analysis

```
> VA1=factanal(fact1, factors=6, rotation='varimax')
> VA1
```

```
> VA1=factanal(fact1, factors=6, rotation='varimax')
> VA1
Call:
factanal(x = fact1, factors = 6, rotation = "varimax")

Uniquenesses:
Domestic_violence      Child_abuse      Economic_problems      Friend_Violence      Academic_stress
          0.873           0.774           0.606           0.005           0.834
School_violence_experience Interested_soldier      Materialism       Games           0.925
                           0.940           0.956           0.945           0.828
                           0.957           0.960           0.904

Loadings:
          Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
Domestic_violence           0.346
Child_abuse                  0.473
Economic_problems            0.622
Friend_Violence              0.995
Academic_stress               0.405
School_violence_experience   0.970
Interested_soldier             0.184
Materialism                   0.290
Games                         0.340  0.236
Internet_addiction            0.107  -0.203
Celebrities                   -0.116 -0.173
Movie                          0.253  0.178

          Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
SS loadings     1.954  0.439  0.363  0.308  0.210  0.113
Proportion Var  0.150  0.034  0.028  0.024  0.016  0.009
Cumulative Var  0.150  0.184  0.212  0.236  0.252  0.261

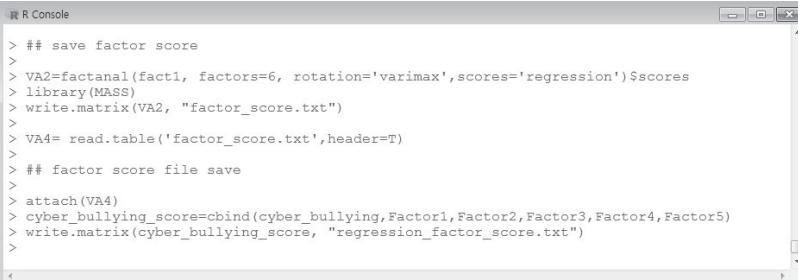
Test of the hypothesis that 5 factors are sufficient.
The chi square statistic is 349.78 on 15 degrees of freedom.
The p-value is 2.33e-65
>
```

Interpretation: In the second-round factor analysis, the explanatory power of Factor 1 was 15.0% (Proportion Var: 0.149), the explanatory power of Factor 2 was 3.4%, the explanatory power of Factor 3 was 2.8%, the explanatory power of Factor 4 was 2.4%, And the explanatory power of factor 6 was 0.009. In this factor analysis, Factor 1 is the school violence factor (School_violence, School_violence_experience), Factor 2 is the economic factor(Economic_problems, Materialism), Factor 3 is the domestic violence factor(Domestic_violence, Child_abuse), Factor 4 is the academic stress factor(Academic_stress, Internet_addiction), Factor 5 is the game factor(Game, Chat_app), and there are no variables included in Factor 6.

★ Save the factor scores.

The factor scores for the six factors derived from the second round of the factor analysis can be stored in a file, which can then be used for running correlation analysis or logistic regressions.

```
> VA2=factanal(fact1, factors=6, rotation='varimax',scores=
  'regression')$scores
  - Assign the factor scores of the six factors derived from the second-
    round factor analysis to the VA2 object.
> library(MASS)
  - Load the "MASS" package, which includes the write.matrix()
    function.
> write.matrix(VA2, "factor_score.txt")
  - Save the factor scores stored in VA2 to the file factor_score.txt.
> VA4= read.table('factor_score.txt',header=T)
  - Assign the data file factor_score.txt to VA4.
> attach(VA4): Attach 'VA4' as execution data.
> cyber_bullying_score=cbind(cyber_bullying,Factor1,Factor2,Factor3,
  Factor4,Factor5)
  - Combine the variables (Factor1–Factor5) containing the factor scores
    derived from the second-round factor analysis and assign this to the
    cyber_bullying_score object.
> write.matrix(cyber_bullying_score, "regression_factor_score.txt")
  - Save the cyber_bullying_score object to the regression_factor_score.txt
    file.
```



```
R Console
> ## save factor score
>
> VA2=factanal(fact1, factors=6, rotation='varimax',scores='regression')$scores
> library(MASS)
> write.matrix(VA2, "factor_score.txt")
>
> VA4= read.table('factor_score.txt',header=T)
>
> ## factor score file save
>
> attach(VA4)
> cyber_bullying_score=cbind(cyber_bullying,Factor1,Factor2,Factor3,Factor4,Factor5)
> write.matrix(cyber_bullying_score, "regression_factor_score.txt")
>
```

★ Conduct binary logistic regression analysis.

```
> regression_factor=read.table(file="regression_factor_score.txt",
  header=T)
> summary(glm(Attitude~Factor1+Factor2+Factor3+Factor4+Factor5,
  family=binomial,data=regression_factor))
  - Run a binary logistic regression.
> exp(coef(glm(Attitude~Factor1+Factor2+Factor3+Factor4+Factor5,
  family=binomial,data=regression_factor)))
  - Compute the odds ratios.
> exp(confint(glm(Attitude~Factor1+Factor2+Factor3+Factor4+Factor5,
  family=binomial,data=regression_factor)))
  - Compute the confidence intervals.
> install.packages('lm.beta')
  - Install the "lm.beta" package for computing standardized regression
    coefficients.
> library(lm.beta)
```

```
R Console

> ## logistic regression : regression_factor_score.txt
>
> regression_factor=read.table(file="regression_factor_score.txt",header=T)
> summary(glm(Attitude~Factor1+Factor2+Factor3+Factor4+Factor5,
+ family=binomial,data=regression_factor))

Call:
glm(formula = Attitude ~ Factor1 + Factor2 + Factor3 + Factor4 +
    Factor5, family = binomial, data = regression_factor)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-1.590 -1.259   1.034   1.098   2.276 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) 0.132644   0.006565 20.205 < 2e-16 ***
Factor1    -0.408230   0.009667 -42.228 < 2e-16 ***
Factor2     0.157321   0.010483 15.007 < 2e-16 ***
Factor3    -0.085365   0.011791 -7.240 4.50e-13 ***
Factor4     0.059587   0.013432  4.436 9.16e-06 ***
Factor5     0.056404   0.016085  3.507 0.000454 *** 
---
Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 136260  on 98682 degrees of freedom
Residual deviance: 133116  on 98677 degrees of freedom
AIC: 133128

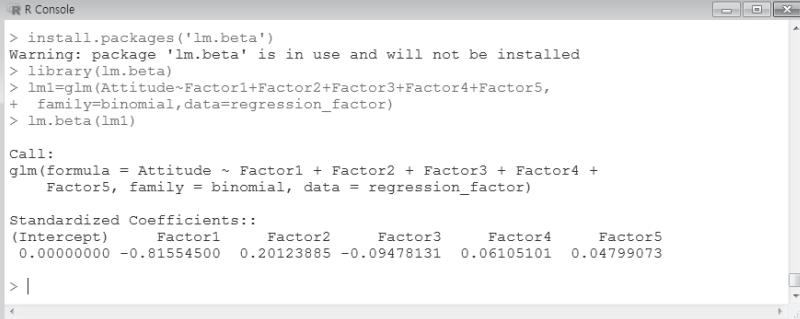
Number of Fisher Scoring iterations: 4

> exp(coef(glm(Attitude~Factor1+Factor2+Factor3+Factor4+Factor5,
+ family=binomial,data=regression_factor)))
(Intercept) Factor1 Factor2 Factor3 Factor4 Factor5
1.1418439 0.6648257 1.1703718 0.9181772 1.0613984 1.0580249
> exp(confint(glm(Attitude~Factor1+Factor2+Factor3+Factor4+Factor5,
+ family=binomial,data=regression_factor)))
Waiting for profiling to be done...
      2.5 %    97.5 %
(Intercept) 1.1272443 1.1566300
Factor1     0.6521953 0.6773907
Factor2     1.1466266 1.1947304
Factor3     0.8971668 0.9396151
Factor4     1.0338740 1.0897773
Factor5     1.0252342 1.0919642
> |
```

Interpretation: Factor 1 (School Violence) and Factor 3 (Domestic Violence) are associated with negative attitudes toward school cyberbullying, while Factor 2 (Economic), Factor 4 (Academic Stress), and Factor 5 (Game) are associated with positive attitudes toward cyber bullying.

★ Compute the standardized regression coefficients

```
> lm1=glm(Attitude~Factor1+Factor2+Factor3+Factor4+Factor5,
family=binomial,data=regression_factor)
> lm.beta(lm1)
- display the standardized regression coefficients on the screen.
```



```

R Console
> install.packages('lm.beta')
Warning: package 'lm.beta' is in use and will not be installed
> library(lm.beta)
> lm1=glm(Attitude~Factor1+Factor2+Factor3+Factor4+
+ family=binomial,data=regression_factor)
> lm.beta(lm1)

Call:
glm(formula = Attitude ~ Factor1 + Factor2 + Factor3 + Factor4 +
    Factor5, family = binomial, data = regression_factor)

Standardized Coefficients:
(Intercept)      Factor1      Factor2      Factor3      Factor4      Factor5
 0.00000000 -0.81554500  0.20123885 -0.09478131  0.06105101  0.04799073

> |

```

Interpretation: In descending order, Factor 1 (School Violence) and Factor 3 (Domestic Violence) are associated with more negative attitudes toward cyber bullying. In descending order, Factor 2 (Economic), Factor 4 (Academic Stress), and Factor 5 (Game) are associated with more positive attitudes toward cyber bullying.

⑩ Reliability Analysis

Reliability refers to the degree to which using the identical or similar measurement instrument (questionnaire) for some subject of measurement (variable) will yield identical or similar results. Thus, reliability is an indication of the consistency among multivariate variables that are measured, and the degree of reliability refers to the variance of the measured values when the same concepts are measured repeatedly. In R, reliability can be measured using Cronbach's alpha coefficient.

Research Question: Assess the reliability of the eight variables included in the GST factors (Strain, Physical, Victim_psychology, Self_control, Attachment, Passion, Offender_psychology, Delinquency).

```

> install.packages('psych')
  - Install the package for conducting reliability analysis.
> library(psych)
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_reliability.txt",
  header=T)
> attach(cyber_bullying)
> factor1=cbind(Strain,Physical,Victim_psychology,Self_control,

```

Attachment,Passion, Offender_psychology,Delinquency)

- Combine the variables to be included in the reliability analysis (Strain~Delinquency) and assign them to factor1.

> alpha(factor1): Compute Cronbach's alpha.

```
R Console
> alpha(factor1)
Warning in alpha(factor1) :
  Some items were negatively correlated with the total scale and probably
should be reversed.
To do this, run the function again with the 'check.keys=TRUE' option
Some items ( Self_control ) were negatively correlated with the total scale and
probably should be reversed.
To do this, run the function again with the 'check.keys=TRUE' option
Reliability analysis
Call: alpha(x = factor1)

  raw_alpha std.alpha G6(smc) average_r S/N    ase mean    sd median_r
  0.41      0.41     0.4      0.079 0.69 0.0022 0.26 0.19     0.099

lower alpha upper      95% confidence boundaries
0.4 0.41 0.41

Reliability if an item is dropped:
   raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
Strain           0.40      0.40     0.39    0.086 0.66 0.0023 0.0097 0.094
Physical         0.38      0.38     0.38    0.081 0.62 0.0024 0.0110 0.105
Victim_psychology 0.34      0.32     0.32    0.063 0.47 0.0025 0.0100 0.045
Self_control     0.47      0.49     0.46    0.119 0.94 0.0021 0.0048 0.119
Attachment        0.33      0.34     0.33    0.067 0.51 0.0026 0.0079 0.067
Passion           0.32      0.33     0.32    0.065 0.49 0.0026 0.0092 0.067
Offender_psychology 0.32      0.30     0.30    0.057 0.42 0.0026 0.0093 0.045
Delinquency       0.43      0.43     0.42    0.096 0.75 0.0021 0.0103 0.119

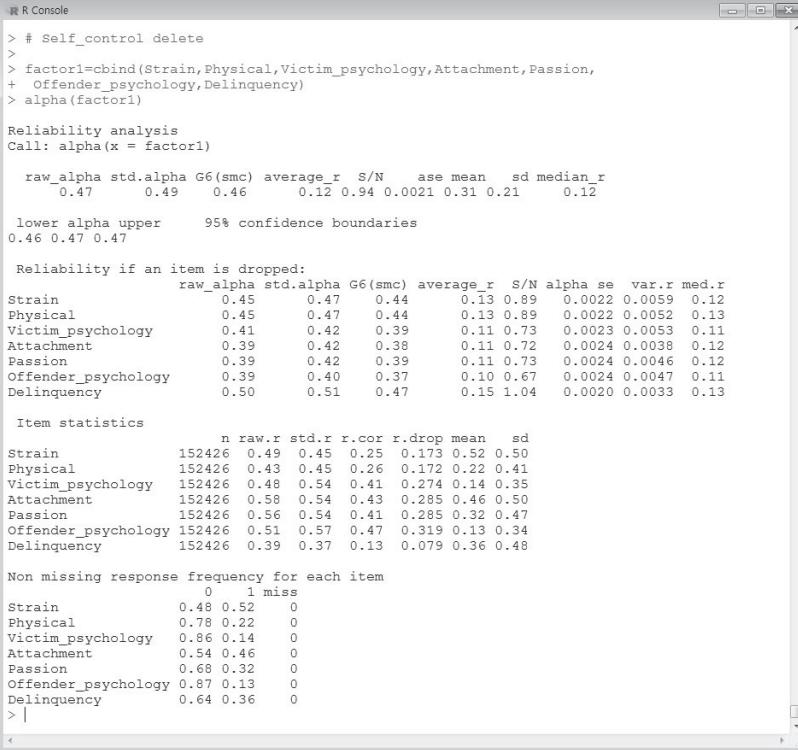
Item statistics
   n raw.r std.r r.cor r.drop mean    sd
Strain      152426 0.46 0.40 0.212 0.140 0.52 0.50
Physical     152426 0.43 0.43 0.253 0.164 0.22 0.41
Victim_psychology 152426 0.49 0.54 0.437 0.280 0.14 0.35
Self_control 152426 0.12 0.21 -0.105 -0.094 0.11 0.32
Attachment    152426 0.56 0.51 0.413 0.262 0.46 0.50
Passion       152426 0.55 0.53 0.426 0.280 0.32 0.47
Offender_psychology 152426 0.52 0.57 0.508 0.329 0.13 0.34
Delinquency   152426 0.38 0.34 0.099 0.061 0.36 0.48

Non missing response frequency for each item
   0    1 miss
Strain 0.48 0.52 0
Physical 0.78 0.22 0
Victim_psychology 0.86 0.14 0
Self_control 0.89 0.11 0
Attachment 0.54 0.46 0
Passion 0.68 0.32 0
Offender_psychology 0.87 0.13 0
Delinquency 0.64 0.36 0
```

Interpretation: The standardized reliability (std.alpha) of the variables included in the GST factors was found to be 0.41. Discarding Self_control resulted in an improved reliability of 0.47. Therefore, a second round of reliability analysis was conducted after discarding Self_control.

★ Second-Round Reliability Analysis

```
> factor1=cbind(Strain,Physical,Victim_psychology,Attachment,Passion,
+ Offender_psychology,Delinquency)
- Combine the variables to be included in the second-round reliability
  analysis (Strain~Delinquency) and assign them to factor1.
> alpha(factor1)
```



The screenshot shows the R console window with the following output:

```

R R Console

> # Self_control delete
>
> factor1=cbind(Strain,Physical,Victim_psychology,Attachment,Passion,
+ Offender_psychology,Delinquency)
> alpha(factor1)

Reliability analysis
Call: alpha(x = factor1)

  raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
 0.47      0.49      0.46      0.12 0.94 0.0021 0.31 0.21      0.12

  lower alpha upper      95% confidence boundaries
 0.46 0.47 0.47

  Reliability if an item is dropped:
  raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
Strain          0.45      0.47      0.44      0.13 0.89 0.0022 0.0059 0.12
Physical        0.45      0.47      0.44      0.13 0.89 0.0022 0.0052 0.13
Victim_psychology 0.41      0.42      0.39      0.11 0.73 0.0023 0.0053 0.11
Attachment      0.39      0.42      0.38      0.11 0.72 0.0024 0.0038 0.12
Passion          0.39      0.42      0.39      0.11 0.73 0.0024 0.0046 0.12
Offender_psychology 0.39      0.40      0.37      0.10 0.67 0.0024 0.0047 0.11
Delinquency      0.50      0.51      0.47      0.15 1.04 0.0020 0.0033 0.13

  Item statistics
  n raw.r std.r r.drop mean   sd
Strain      152426 0.49 0.45 0.25 0.173 0.52 0.50
Physical     152426 0.43 0.45 0.26 0.172 0.22 0.41
Victim_psychology 152426 0.48 0.54 0.41 0.274 0.14 0.35
Attachment    152426 0.58 0.54 0.43 0.285 0.46 0.50
Passion        152426 0.56 0.54 0.41 0.285 0.32 0.47
Offender_psychology 152426 0.51 0.57 0.47 0.319 0.13 0.34
Delinquency    152426 0.39 0.37 0.13 0.079 0.36 0.48

  Non missing response frequency for each item
  0    1 miss
Strain      0.48 0.52 0
Physical     0.78 0.22 0
Victim_psychology 0.86 0.14 0
Attachment    0.54 0.46 0
Passion        0.68 0.32 0
Offender_psychology 0.87 0.13 0
Delinquency    0.64 0.36 0
> |
```

Interpretation: The results of the second-round analysis indicate that the standardized reliability (std.alpha) of the variables included in the GST factors is 0.49.

⑦ Multivariate Analysis of Variance

In two-way ANOVA, the researcher analyzes one dependent variable (the one-week spread: Onespread) and two independent variables (Account, Channel), testing for differences in the means of the dependent variable across different groups. In multivariate ANOVA, or MANOVA, two or more dependent variables and two or more independent variables are analyzed, to examine the differences in the means of the dependent variables across different groups.

Research Question: Are there differences in the means of the dependent variables (Onespread, Twospread) across the independent variables relevant to cyber bullying (Account, Channel)?

```
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_
analysis.txt", header=T)
> attach(cyber_bullying)
> tapply(Onespread, Channel, mean)
  - Compute the mean of Onespread by Channel.
> tapply(Onespread, Channel, sd)
  - Compute the standard deviation of Onespread by Channel.
> tapply(Onespread, Account, mean)
  - Compute the mean of Onespread by Account.
> tapply(Onespread, Account, sd)
  - Compute the standard deviation of Onespread by Account.
> tapply(Onespread, list(Channel,Account), mean)
  - Compute the mean of Onespread by Channel and Account.
> tapply(Onespread, list(Channel,Account), sd)
  - Compute the standard deviation of Onespread by Channel and
  Account.
> tapply(Twospread, Channel, mean)
> tapply(Twospread, Channel, sd)
> tapply(Twospread, Account, mean)
> tapply(Twospread, Account, sd)
> tapply(Twospread, list(Channel,Account), mean)
> tapply(Twospread, list(Channel,Account), sd)
```

```

R Console
> ## multivariate analysis of variance (MANOVA)
>
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_descriptive_analysis.txt",header=T)
> #attach(cyber_bullying)
>
> tapply(Onespread, Channel, mean)
      1       2       3       4       5
169.9582487 0.9420638 1.7308564 0.1899315 5.6873034
> #tapply(Onespread, Channel, sd)
> tapply(Onespread, Account, mean)
      0       1
0.6178124 181.4953337
> #tapply(Onespread, Account, sd)
> tapply(Onespread, list(Channel,Account), mean)
      0       1
1 1.63075209 269.716521
2 0.21703177 5.011808
3 0.87185421 4.833856
4 0.06734007 3.107372
5 1.48432056 10.655409
> #tapply(Onespread, list(Channel,Account), sd)
> tapply(Twospread, Channel, mean)
      1       2       3       4       5
2.16244455 0.06661741 0.07227005 0.02042135 0.11778029
> #tapply(Twospread, Channel, sd)
> tapply(Twospread, Account, mean)
      0       1
0.009212527 2.383166104
> #tapply(Twospread, Account, sd)
> tapply(Twospread, list(Channel,Account), mean)
      0       1
1 0.011810585 3.4370048
2 0.012186276 0.3721498
3 0.013017067 0.2863114
4 0.002760943 0.4407051
5 0.013472706 0.2410763
> #tapply(Twospread, list(Channel,Account), sd)

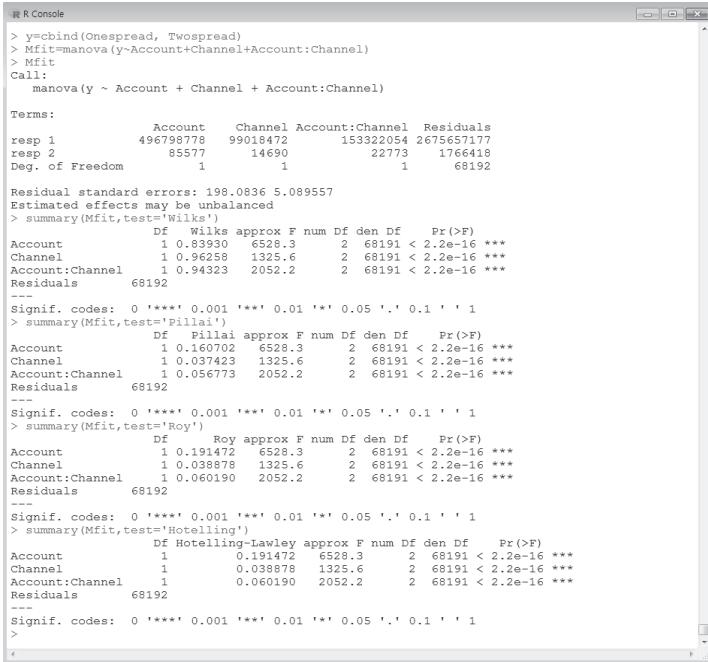
```

Interpretation: Looking at the descriptive statistics computed above, to compare the means of Onespread and Twospread depending on Account (First, Spread) and Channel (Twitter, Blog, Cafe, Board, News), the means of Onespread and Twospread of Twitter documents were found to be higher in the case of spread documents (Spread) compared to first postings (First).

★ Multivariate ANOVA

- > y=cbind(Onespread, Twospread)
 - Assign the dependent variables to the y vector.
- > Mfit=manova(y~Account+Channel+Account:Channel)
 - Run two-way MANOVA.
 - y: dependent variable.
 - Account: analysis of the effects of the independent variable (Account)
 - Channel: analysis of the effects of the independent variable (Channel)
 - Account:Channel: analysis of interaction effects between Account and Channel.

> Mfit: Display the output of the two-way MANOVA on the screen.
 > summary(Mfit,test='Wilks')
 - Display the output of the Wilks multivariate test on the screen.
 > summary(Mfit,test='Pillai')
 - Display the output of the Pillai multivariate test on the screen.
 > summary(Mfit,test='Roy')
 - Display the output of the Roy multivariate test on the screen.
 > summary(Mfit,test='Hotelling')
 - Display the output of the Hotelling multivariate test on the screen.



```

R R Console
> yychind(Onespread, Twospread)
> Mfit=manova(y~Account+Channel+Account:Channel)
> Mfit
Call:
  manova(y ~ Account + Channel + Account:Channel)

Terms:
          Account    Channel Account:Channel Residuals
resp 1      496798778  99018472   153322054 2675657177
resp 2       85577    14690      22773     1766418
Deg. of Freedom 1           1           1      68192

Residual standard errors: 198.0836 5.089557
Estimated effects may be unbalanced
> summary(Mfit,test='Wilks')
   Df Wilks approx F num Df den Df Pr(>F)
Account      1 0.83930 6528.3      2 68191 < 2.2e-16 ***
Channel      1 0.96258 1325.6      2 68191 < 2.2e-16 ***
Account:Channel 1 0.94323 2052.2      2 68191 < 2.2e-16 ***
Residuals    68192
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
> summary(Mfit,test='Pillai')
   Df Pillai approx F num Df den Df Pr(>F)
Account      1 0.160702 6528.3      2 68191 < 2.2e-16 ***
Channel      1 0.037423 1325.6      2 68191 < 2.2e-16 ***
Account:Channel 1 0.056773 2052.2      2 68191 < 2.2e-16 ***
Residuals    68192
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
> summary(Mfit,test='Roy')
   Df Roy approx F num Df den Df Pr(>F)
Account      1 0.191472 6528.3      2 68191 < 2.2e-16 ***
Channel      1 0.038878 1325.6      2 68191 < 2.2e-16 ***
Account:Channel 1 0.060190 2052.2      2 68191 < 2.2e-16 ***
Residuals    68192
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
> summary(Mfit,test='Hotelling')
   Df Hotelling-Lawley approx F num Df den Df Pr(>F)
Account      1 0.191472 6528.3      2 68191 < 2.2e-16 ***
Channel      1 0.038878 1325.6      2 68191 < 2.2e-16 ***
Account:Channel 1 0.060190 2052.2      2 68191 < 2.2e-16 ***
Residuals    68192
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
>

```

Interpretation: The results of the above multivariate tests indicate that there are significant differences in the means of Onespread and Twospread across Account (Wilks lambda = .839, $p <.001$) and Channel (Wilks lambda = .962, $p <.001$). The interaction effect test indicates that the Wilks lambda of Account×Channel is .943 with $F = 2052.2$, showing that there is a significant difference ($p <.001$), and thus rejecting the null hypothesis of ‘no interaction effect’. The highest means of Onespread and Twospread were observed in spread documents on Twitter.

★ Between-subjects Effect Test

> summary.aov(Mfit)

```
R Console
> ## between-subjects effect test
>
> summary.aov(Mfit)
Response Onespread :
  Df   Sum Sq  Mean Sq F value    Pr(>F)
Account        1 496798778 496798778 12661.5 < 2.2e-16 ***
Channel        1  99018472  99018472  2523.6 < 2.2e-16 ***
Account:Channel 1 153322054 153322054  3907.6 < 2.2e-16 ***
Residuals     68192 2675657177   39237
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Response Twospread :
  Df   Sum Sq  Mean Sq F value    Pr(>F)
Account        1  85577   85577 3303.66 < 2.2e-16 ***
Channel        1  14690   14690  567.09 < 2.2e-16 ***
Account:Channel 1  22773   22773  879.15 < 2.2e-16 ***
Residuals     68192 1766418      26
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Interpretation: The results of the between-subjects effect test indicate that there are significant differences in the means of Onespread ($F = 12661.5, p <.001$) and Twospread ($F = 3303.7, p <.001$) depending on the value of Account. Likewise, there are significant differences in the means of Onespread ($F = 2523.6, p <.001$) and Twospread ($F = 567.09, p <.001$) depending on the type of Channel. Significant differences were also found in the means of Onespread ($F = 3907.6, p <.001$) and Twospread ($F = 879.2, p <.001$) depending on the value of Account \times Channel.

⑯ Binary Logistic Regression Analysis

Binary, or dichotomous, logistic regression analysis examines the effect of quantitative independent variables on a dependent variable that takes binary (0, 1) values.

Research Question: Which are the GST factors (Strain~Delinquency) affecting attitudes to cyber bullying [Attitude (Negative, Positive)]?

```
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_methodology_
  numeric.txt", header=T)
```

```
> input=read.table('input_binary_logistic.txt',header=T,sep=",")  
  - Assign the independent variables to the input object, using the (,) delimiter.  
> output=read.table('output_binary_logistic.txt',header=T,sep=",")  
  - Assign the dependent variable to the output object, using the (,) delimiter.  
> attach(cyber_bullying)  
> input_vars = c(colnames(input))  
  - Assign the input variables as a vector value to the input_vars variable.  
> output_vars = c(colnames(output))  
  - Assign the output variable as a vector value to the output_vars variable.  
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',  
  paste(input_vars, collapse = '+')))  
  - Using the paste function to concatenate character strings, assign the functional form of the logistic regression model to the form variable.  
> form: Display the functional form of the logistic regression equation.  
> summary(glm(form, family=binomial,data=cyber_bullying))  
  - Run the binary logistic regression.  
> exp(coef(glm(form, family=binomial,data=cyber_bullying)))  
  - Compute the odds ratios.  
> exp(confint(glm(form, family=binomial,data=cyber_bullying)))  
  - Compute the confidence intervals.
```

```

R Console
> summary(glm(form, family=binomial,data=cyber_bullying))

Call:
glm(formula = form, family = binomial, data = cyber_bullying)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-2.5935 -0.9738  0.6034  0.8986  2.1491 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -0.21808  0.01114 -19.58 <2e-16 ***
Strain       -0.28179  0.01167 -24.14 <2e-16 ***
Physical     -0.18361  0.01395 -13.16 <2e-16 ***
Victim_psychology -0.61098  0.01719 -35.54 <2e-16 ***
Self_control  1.43496  0.02084  68.86 <2e-16 ***
Attachment   1.25221  0.01213 103.26 <2e-16 ***
Passion      0.85885  0.01289  66.64 <2e-16 ***
Offender_psychology -0.18098  0.01796 -10.08 <2e-16 ***
Delinquency   -0.72930  0.01198 -60.89 <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: 209971  on 152425  degrees of freedom
Residual deviance: 181309  on 152417  degrees of freedom
AIC: 181327

Number of Fisher Scoring iterations: 4

> exp(coef(glm(form, family=binomial,data=cyber_bullying)))
             Strain          Physical        Victim_psychology      Self_control  
 0.8040608      0.7544295      0.8322572      0.5428178      4.1994639 
  Attachment      Passion Offender_psychology      Delinquency  
 3.4980603      2.3604436      0.8344529      0.4822458  
> exp(confint(glm(form, family=binomial,data=cyber_bullying)))
Waiting for profiling to be done...
              2.5 %    97.5 %    
(Intercept) 0.7866936 0.8218004
Strain      0.7373642 0.7718833
Physical    0.8098080 0.8553254
Victim_psychology 0.5248225 0.5614088
Self_control 4.0318750 4.3750695
Attachment   3.4159561 3.5822587
Passion      2.3016019 2.4208722
Offender_psychology 0.8055880 0.8643527
Delinquency  0.4710530 0.4936953
>

```

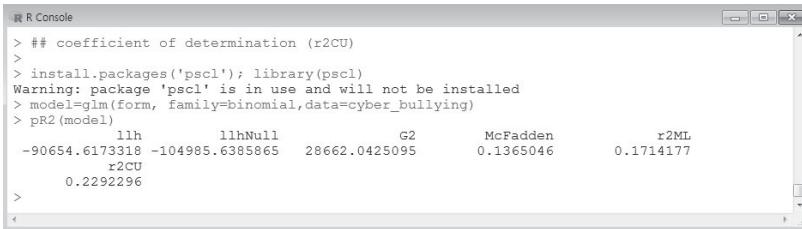
Interpretation: Strain, Physical, Victim_psychology, Offender_psychology, and Delinquency have a negative effect on cyber bullying, while Self_control, Attachment, and Passion have a positive effect on cyber bullying.

- ★ Compute the coefficient of determination of the binary logistic regression model

```

> install.packages('pscl'); library(pscl)
> model=glm(form, family=binomial,data=cyber_bullying)
> pR2(model)

```



```

R Console
> ## coefficient of determination (r2CU)
>
> install.packages('pscl'); library(pscl)
Warning: package 'pscl' is in use and will not be installed
> model<-glm(form, family=binomial,data=cyber_bullying)
> pR2(model)
      1lh      1lhNull       G2      McFadden      r2ML
-90654.6173318 -104985.6305865  28662.0425095     0.1365046   0.1714177
      r2CU
  0.2292296
>

```

Interpretation: The coefficient of determination (r2CU) is computed to be 0.229, indicating that the estimated logistic regression model explains about 22.9% of the variation within the dataset.

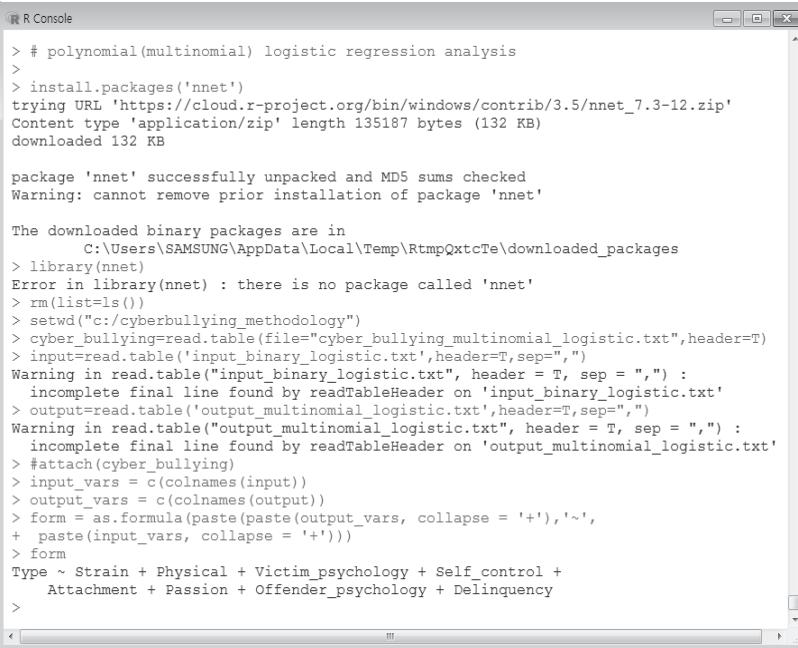
⑯ Multinomial Logistic Regression Analysis

Multinomial, or polychotomous, logistic regression analysis examines the effects of quantitative independent variables on a dependent variable that takes at least three different categories of values [type (1 = offender, 2 = victim, 3 = bystander, 4 = complex)].

```

> install.packages('nnet')
- The “nnet” package includes the multinom() function, which is used
  for running multinomial logistic regressions.
> library(nnet)
> library(nnet)
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_multinomial_
  logistic.txt", header=T)
> input=read.table('input_binary_logistic.txt',header=T,sep=",")
> output=read.table('output_multinomial_logistic.txt',header=T,sep=",")
> attach(cyber_bullying)
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
> form

```



The screenshot shows the R Console window with the following R script:

```

R Console

> # polynomial(multinomial) logistic regression analysis
>
> install.packages('nnet')
trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.5/nnet_7.3-12.zip'
Content type 'application/zip' length 135187 bytes (132 KB)
downloaded 132 KB

package 'nnet' successfully unpacked and MD5 sums checked
Warning: cannot remove prior installation of package 'nnet'

The downloaded binary packages are in
  C:\Users\SAMSUNG\AppData\Local\Temp\RtmpQxtcTe\downloaded_packages
> library(nnet)
Error in library(nnet) : there is no package called 'nnet'
> rm(list=ls())
> setwd("c:/cyberbullying_methodology")
> cyber_bullying=read.table(file="cyber_bullying_multinomial_logistic.txt",header=T)
> input=read.table('input_binary_logistic.txt',header=T,sep=",")
Warning in read.table("input_binary_logistic.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_binary_logistic.txt'
> output=read.table('output_multinomial_logistic.txt',header=T,sep=",")
Warning in read.table("output_multinomial_logistic.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_multinomial_logistic.txt'
> #attach(cyber_bullying)
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>

```

> model=multinom(form,data=cyber_bullying)
 - Run the multinomial logistic regression.

> summary(model)
 - Print the output of the multinomial logistic regression to the screen.

> z=summary(model)\$coefficients/summary(model)\$standard.errors
 - As the multinom function cannot be used to obtain p-values, we use
 the z-tests (Wald tests) to compute the p-values.

> p=(1-pnorm(abs(z), 0, 1))*2: Compute the p-values.

> p: Display the p-values on the screen.

> exp(coef(model))
> exp(confint(model))

```

R R Console
> model=multinom(form,data=cyber_bullying)
# weights: 40 (27 variables)
initial value 82722.957117
iter 10 value 73392.284410
iter 20 value 65307.086888
iter 30 value 65308.495948
iter 40 value 53154.419871
final value 53154.358454
converged
> summary(model)
Call:
multinom(formula = form, data = cyber_bullying)

Coefficients:
(Intercept) Strain Physical Victim_psychology Self_control Attachment Passion Offender_psychology Delinquency
2 1.9015719 -0.7795003 0.3393008 0.5063166 0.2400371 0.4574945 0.3055380 0.5714538 -0.4435724
3 -0.3271234 -0.6276526 -0.1208946 0.2431184 0.3295161 0.5079155 0.8169261 0.6571957 -0.3374535
4 -0.5912969 -0.6587167 0.1072804 0.5137968 0.5562438 0.6798739 0.5787695 0.7484835 -0.5087588

Std. Errors:
(Intercept) Strain Physical Victim_psychology Self_control Attachment Passion Offender_psychology Delinquency
2 0.02954138 0.02875530 0.03237844 0.04043435 0.06743862 0.02989307 0.03240989 0.04611781 0.02658422
3 0.04001094 0.03905142 0.04459926 0.05262018 0.08247812 0.03987155 0.04207905 0.05694285 0.03708751
4 0.04236430 0.04141693 0.04546575 0.05331131 0.08259812 0.04248887 0.04478165 0.05659195 0.03993386

Residual Deviance: 106308.7
AIC: 106362.7
> z=summary(model)$coefficients/summary(model)$standard.errors
> p=(1-pnorm(abs(z), 0, 1))*2
> p
(Intercept) Strain Physical Victim_psychology Self_control Attachment Passion Offender_psychology Delinquency
2 0.000000e+00 0 0.000000000 0.000000e+00 3.717852e-04 0 0 0 0
3 2.20446e-16 0 0.006714438 3.832789e-06 6.464140e-05 0 0 0 0
4 0.000000e+00 0 0.018295291 0.000000e+00 1.646749e-11 0 0 0 0
> |

```

Interpretation: The results of the multinomial logistic regression model for examining the GST factors influencing the types of involvement with school cyberbullying are as follows. The Strain and Delinquency factors were found to affect offenders (1: reference category) more than victims (2), bystanders (3), and complex types (4). The Physical factor was found to affect victims (2) and complex types (4) more than offenders (1), and offenders (1) more than bystanders (3). The Victim_psychology, Self_control, Attachment, Passion, and Offender_psychology factors were found to affect victims (2), bystanders (3), and complex types (4) more than offenders (1).

References

- Baron, R. M. & Kenny, D. A. (1986). The moderator-mediator variable in social psychological research: conceptual, strategic, and statistics considerations. *Journal of Personality and Social Psychology*, 51(6), 1173-1182.
- Kline, R. B. (2010). Principles and Practice of Structural Equation Modeling(3rd ed.). NY: Guilford Press.
- Montgomery, Douglas C. & Runger, George C. (2003). Applied Statistics and Probability for Engineers. John Wiley & Sons, Inc.
- Song, J.Y., Song, T.M., Seo, D.C., Jin, J.H. (2016). Data Mining of Web-Based Documents on Social Networking Sites That Included Suicide-Related Words Among Korean Adolescents. *Journal of Adolescent Health*, 59 (2016), 668-673.

OVERVIEW OF MACHINE LEARNING

Introduction

Machine learning is a computer-science field that uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) with data, without being explicitly programmed (https://en.wikipedia.org/wiki/Machine_Learning (accessed on 13 May 2018)). Artificial intelligence (AI), also known as machine intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans. In computer science, AI research is defined as the study of "intelligent agents": any device that perceives its environment and takes actions maximizing chances of successfully achieving its goals. The term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, e.g., "learning" and "problem solving" (https://en.wikipedia.org/wiki/Artificial_intelligence (accessed on 13 May 2018)).

Data mining is related to machine learning and can be defined as "the science of extracting useful information from large data sets or databases" (Hand et al., 2001: p. 2). Data mining can be applied to a variety of fields (classification, clustering, association, sequencing, forecasting, etc.) as a data-analysis method for finding results. Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader concept of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Most modern deep-learning models are based on an artificial neural network (https://en.wikipedia.org/wiki/Deep_learning (accessed on 13 May 2018)).

The goal of machine learning is to learn by using existing data and then to find predictive values (i.e., labels) of new data, based on features that were found through learning. That is, machine learning refers to algorithms that learn on their own, based on probability and data, to infer results. On the other hand, the goal of data mining is to discover properties of existing data that were previously unknown and to find statistical rules or patterns. Machine learning and data mining are both based on data and are employed together to solve problems, using technology such as classification, forecasting, clustering, models, algorithms, etc. (Fig. 1).

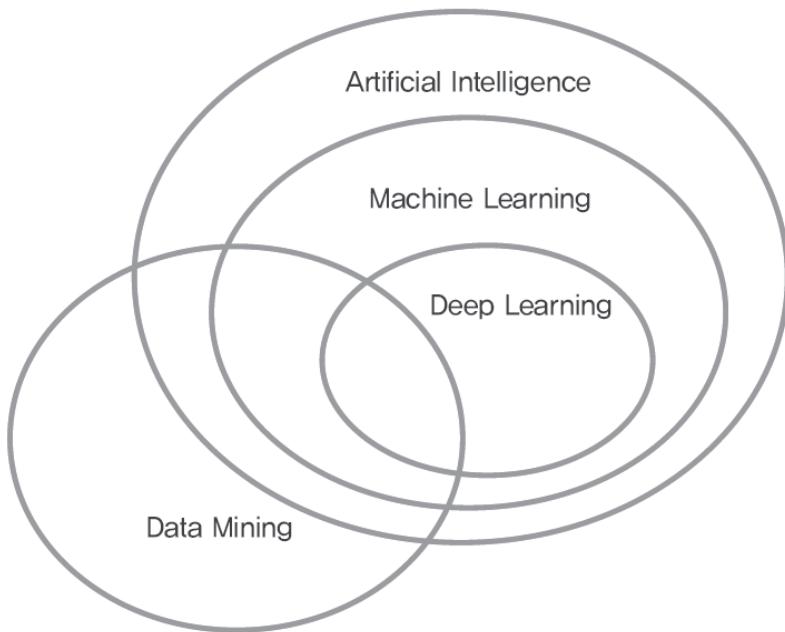


Fig. 1. AI, Machine Learning, Deep Learning, and Data Mining

The learning methods used in machine learning¹ can be broadly divided into supervised learning, unsupervised learning, and reinforcement learning. Supervised learning learns with both independent variables and dependent variables in circumstances where the training data contains independent variables (feature vectors) and dependent variables (labels). A trained predictive model receives, as an input, new data that does not contain dependent variables. The model extracts the predicted dependent variables (expected labels) from only the independent variables included in the new data (Fig. 2). Machine learning algorithms in the supervised-learning category include the Naïve Bayes classification model, logistic regression model, random forest model, decision tree model, neural network model, and support vector machine model.

Unsupervised learning learns with only independent variables in a situation with no dependent variables in the training data. A trained

¹ Machine Learning Terminology: Features are data properties; Feature Vectors are Independent Variables; A label classifies data; Labels are Dependent Variables; Training Data are data that include Feature Vectors and Labels.

predictive model receives, as an input, new data that does not contain dependent variables. The model extracts the predicted dependent variables (expected labels) from only the independent variables included in the new data (Fig. 3). Unsupervised learning models are used in clustering analysis and association analysis.

Reinforcement learning models the process of receiving rewards through trial and error and learning behavior patterns. In reinforcement learning, an agent receives input in the current state and learns to select from among the rules that have been created. Then, the agent can receive rewards from the environment by performing actions in the environment. Thus, the learning algorithm is repeatedly updated (Fig. 4).

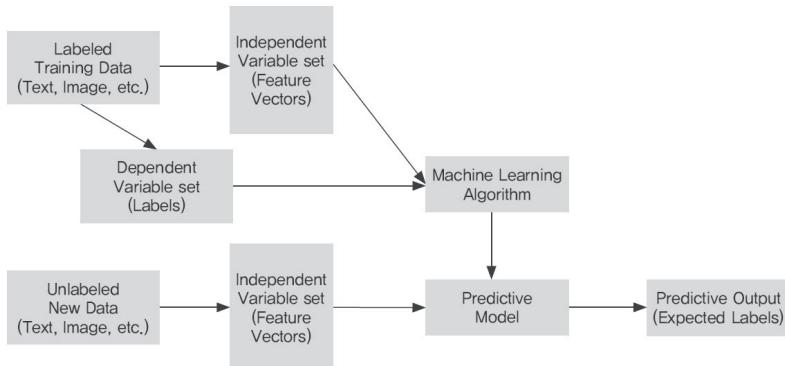


Fig. 2. Supervised Learning Modeling

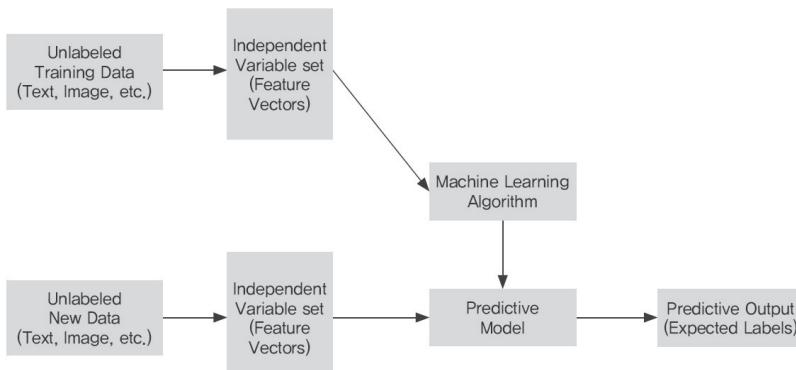
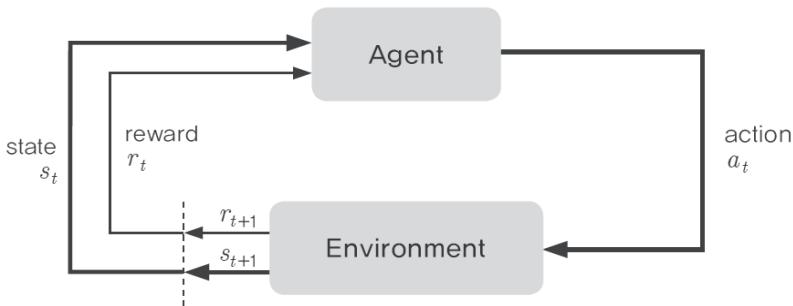


Fig. 3. Unsupervised Learning Modeling



Reference: <https://www.analyticsvidhya.com/blog/2017/01/introduction-to-reinforcement-learning-implementation/>

Fig. 4. Reinforcement Learning Modeling

The advantages of machine learning are as follows. First, when the machine learning algorithm is used properly, forecasting performance can be superior to traditional parametric modeling such as logistic regression analysis. In particular, it can be more useful when there is a complex nonlinear relationship between predictors and outcomes of interest (Berk & Bleich, 2014). Second, the process of machine learning does not need to worry about specifics on how researchers will tune variables to achieve optimal performance for each research topic. Therefore, machine learning algorithms minimize interaction with humans in predicting output variables (Duwe & Kim, 2017).

To use machine learning to develop a prediction model, the following things must be considered. First, the scale of the input variables (independent variables) and output variables (dependent variables) must be determined. The scale assigns values according to the qualitative state of the properties of the object of observation; it can be broadly classified as categorical data or continuous data. When determining the scale of the input and output variables, the scale's categories must appear at a sufficient frequency to be able to use machine learning. For example, if a variable's scale is continuous and an insufficient sample size is extracted from the population, or the frequency of each of the variable's categories is insufficient, the machine learning prediction results may show that the prediction model's performance has greatly decreased because of the low probability of the category appearing.

When applying the continuous independent variable to the machine learning algorithm, it can affect the prediction of the dependent variable more than the other categorical independent variables used in the learning. On the other hand, if continuous independent variables are transformed

into categorical variables and applied to machine learning algorithms, it is possible to estimate the probability that the group contributes to predicting the dependent variable, although it can bring the loss of information due to grouping.

Therefore, when examining input and output variable categories, their appearance frequency should be considered to determine a categorical scale. Second, the number of input variables must be considered. With a large number of input variables, there is a risk of unbalanced data; i.e., if there is a relatively small amount of data for a particular variable, the prediction performance may suffer.

If there is a lot of missing in the input variable, learning data can be constructed by converting missing to dummy variable without replacing missing with measured value. In order to develop artificial intelligence through machine learning, learning data is divided into training data and test data. Then, a model is developed with training data and evaluated with test data.

This chapter discusses supervised-learning analysis technologies for developing prediction models using machine learning, including the Naïve Bayes classification model, logistic regression model, random forest model, decision tree model, neural network model, and support vector machine model. It also discusses unsupervised learning technologies, e.g., association analysis, cluster analysis, evaluation of machine learning models, and visualization.

Machine Learning Training Data

As seen in the previous chapter, training data on the relevant topic is needed for machine learning. The training data used for machine learning in this study included 350,314 cyber bullying online documents collected from Korean online channels from January 1, 2013 to June 30, 2017. The main items used in this study's training data are shown in Table 1. The time variables (Year, Month, Day, Hour, Week) and region variables were used as the items for the documents' ID (identification).

For the documents' dependent variables (labels), emotions toward cyber bullying (Negative, Positive) and the type (Perpetrator, Victim, Bystander, Complex) were used. For the documents' independent variables (feature vectors), this study used the strain factors that are the major factors in Agnew's general strain theory (GST). For the data used in the association analysis, 12 delinquency factors were used.

Table 1 Main items in machine learning training data files

Item	Variables	Contents
ID	Year	2013-2017
	Month	1-12
	Day	1-31
	Hour	1-24
	Week	Monday - Sunday
	Region	Seoul-Jeju
cyber bullying emotion	Attitude	0(Neutral+Negative): Negative, 1: Positive
	Positive	0: No, 1: Yes
	Negative	0: No, 1: Yes
cyber bullying type	Type	1: Perpetrator, 2: Victim, 3: Bystander 4: Complex 5: Non involved
	Perpetrator	0: No, 1: Yes
	Victim	0: No, 1: Yes
	Bystander	0: No, 1: Yes
	Complex	0: No, 1: Yes
	Non involved	0: No, 1: Yes
GST factors	Strain	0: No, 1: Yes
	Physical	0: No, 1: Yes
	Victim psychology	0: No, 1: Yes
	Self control	0: No, 1: Yes
	Attachment	0: No, 1: Yes
	Passion	0: No, 1: Yes
	Offender psychology	0: No, 1: Yes
	Delinquency	0: No, 1: Yes
	Access entertainment facilities	0: No, 1: Yes
	Smoking	0: No, 1: Yes
Delinquency factors	Drinking	0: No, 1: Yes
	Drug	0: No, 1: Yes
	Run away	0: No, 1: Yes
	Gambling	0: No, 1: Yes
	Crime	0: No, 1: Yes
	Pregnant	0: No, 1: Yes
	Sexual violence	0: No, 1: Yes
	Sex	0: No, 1: Yes
	Absence without leave	0: No, 1: Yes
	Student violence	0: No, 1: Yes

DEVELOPMENT OF A CYBER BULLYING PREDICTION MODEL BASED ON MACHINE LEARNING

This chapter discusses supervised learning analysis technologies for developing machine learning prediction models, including the Naïve Bayes classification model, logistic regression model, random forest model, decision tree model, neural network model, and support vector machine model. Then, prediction models for cyber bullying are developed using these models.

Naïve Bayes Classification Model

The Naïve Bayes classification model refers to a classifier or learning method that is based on Bayes' theorem, which is a rule regarding conditional probability. Bayes' theorem $[p(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(B|A) \times P(A)}{P(B)}]$ states that in a prior probability, when a certain event occurs, its probability can change. That is, classification models can predict based on the fact that a posterior probability can make predictions via prior probability. Here, $P(A|B)$ is the probability that A will happen if B has happened. $P(B|A)$ is the probability that B will happen if A has happened. $P(A, B)$ is the probability that A and B will happen simultaneously, $P(A)$ is the probability that A will happen, and $P(B)$ is the probability that B will happen.

In Naïve Bayes, it is assumed that the features used in the classifiers are stochastically independent of each other to make classification quick and easy. However, errors can occur if the assumption of stochastic independence is incorrect. If the correlations between properties are considered for data that has many properties, the process becomes complex. Naïve Bayes simplifies this and is used when making rapid judgments, e.g., real-time predictions. It is often used in the fields of spam-mail classification and disease forecasting. For example, Table 2 shows whether or not a game is played, according to the weather outlook. The probability of the outlook (B) being sunny when the game is being played (A) is calculated in terms of conditional probability, using the following formula:

$$\begin{aligned} P(B|A) &= \frac{P(B \cap A)}{P(A)} = P(\text{outlook} = \text{sunny} | \text{play} = \text{yes}) \\ &= \frac{P(\text{outlook} = \text{sunny} \cap \text{play} = \text{yes})}{P(\text{play} = \text{yes})} = \frac{\frac{2}{14}}{\frac{9}{14}} = \frac{2}{9} \end{aligned}$$

If Naïve Bayes classification $[p(A|B) = \frac{P(B|A) \times P(A)}{P(B)}]$ is applied to the probability that a game is being played (A) when the outlook (B) is sunny, we obtain the following formula:

$$\begin{aligned} P(\text{play} = \text{yes} | \text{outlook} = \text{sunny}) &= \frac{P(\text{outlook} = \text{sunny} | \text{play} = \text{yes})P(\text{play} = \text{yes})}{P(\text{outlook} = \text{sunny})} \\ &= \frac{\frac{2}{9} \times \frac{9}{14}}{\frac{5}{14}} = \frac{2}{5} \end{aligned}$$

The advantages of Naïve Bayes are as follows. First, it can train data very effectively in a supervised learning environment, and can be used even if very little training data are available for estimating the parameters needed for classification. Second, it is possible to perform predictions quickly and easily in multi-class, which have several categories.

The disadvantages are as follows. First, categories that are not in the training data but are in the test data have a probability of 0; thus, a zero frequency occurs and makes normal prediction impossible. To resolve this problem, the Laplace Smoothing method is used, which adds 1 to each numerator. Second, errors can occur if the assumption of stochastic independence is incorrect.

Table 2 Whether a game is played, according to the weather outlook

outlook(B)	play(A)
rainy	no
rainy	no
sunny	no
sunny	no
sunny	no
overcast	yes

rainy	yes
rainy	yes
rainy	yes
sunny	yes
sunny	yes

Reference: Mitchell, Tom. M. 1997. Machine Learning. New York: McGraw-Hill., p. 59.

Cyber bullying Risk (Negative, Positive) Prediction Model

To develop a Naïve Bayes classification model that predicts emotions (Negative, Positive) about cyber bullying, a total of 152,426 data values [Negative: 69,083 (45.3%), Positive: 83,343 (54.7%)], which mention emotions about cyber bullying and have the frequency of GST factors, were used as training data. In R, David Meyer's "e1071" package was used for Naïve Bayes classification.

```
> rm(list=ls()): Initialize all variables.
> setwd("c:/cyberbullying_2017"): Set the working directory.
> install.packages('MASS'): Install the MASS packages.
> library(MASS)
  - Load the MASS package containing the write.matrix () function
> install.packages('e1071'): Install the 'e1071' packages.
> library(e1071): Load the 'e1071' packages.
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
  - Assign a training data file to the tdata object.
  - To develop a predictive model (model function) as a supervised
    learning, the category of the attitude included in the training data
    should be coded in numeric format (Negative = 0, Positive = 1).
> input=read.table('input_GST.txt',header=T,sep=",")
  - Assign an independent variable to the input object as a delimiter (,).
> output=read.table('output_attitude.txt',header=T,sep=",")
  - Assign an dependent variable to the output object as a delimiter (,).
> p_output=read.table('p_output_bayes.txt',header=T,sep=",")
  - The predicted value of the Bayes model is assigned to the p_output
    object as a delimiter (,).
> input_vars = c(colnames(input))
  - Assign the input variable to the 'input_vars' variable as a vector
    value.
> output_vars = c(colnames(output))
  - Assign the output variable to the 'output_vars' variable as a vector
    value.
```

```
> p_output_vars = c(colnames(p_output))
  - Assign the 'p_output' variable as a vector value to the 'p_output_vars' variable.
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
  - Assign a function expresses of the Naïve Bayes Model to form the variable using a function that combines strings (paste).
> form: display the function express of the Naïve Bayes Model.
> train_data.lda=naiveBayes(form,data=tdata)
  - Create a model function (classifier) by executing Naive Bayes Classification model with the tdata set.
  - In the training data, there is zero probability in the category of test data. To solve this problem, please use laplace smoothing method which gives +1 for each numerator.
  - train_data.lda=naiveBayes(form,data=tdata, laplace=1).
> p=predict(train_data.lda, tdata, type='raw')
  - Perform model predictions with the tdata and then create the risk prediction group(the dependent variable predicted only by independent variables of the tdata set).
> dimnames(p)=list(NULL,c(p_output_vars))
  - Assign the probability value of the predicted dependent variable to posterior.0 (negative predictive probability) and posterior.1 (positive predictive probability) variable.
> summary(p)
  - Dispaly descriptive statistics of predicted probability value of dependent variable (negative, positive) on the screen.
> pred_obs = cbind(tdata, p)
  - Append the 'posterior.0' and 'posterior.1' variables to the 'tdata data set' and assign them to the 'pred_obs' object.
> write.matrix(pred_obs,'cyberbullying_attitude_naive.txt')
  - Save the pred_obs object as 'cyberbullying_attitude_naive.txt' file.
> m_data = read.table('cyberbullying_attitude_naive.txt',header=T)
  - Assign the 'cyberbullying_attitude_naive.txt' file to the m_data object.
> attach(m_data): Attach m_data to the execution data.
> mean(m_data$posterior.0): Display the negative prediction probability to the screen.
> mean(m_data$posterior.1): Display the positive prediction probability to the screen.
```

```

R Console

> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
> p_output=read.table('p_output_bayes.txt',header=T,sep=",")
Warning message:
In read.table("p_output_bayes.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_bayes.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
> train_data.lda=naiveBayes(form,data=tdata)
> #train_data.lda=naiveBayes(form,data=tdata, laplace=1)
> p=predict(train_data.lda, tdata, type='raw')
>
> dimnames(p)=list(NULL,c(p_output_vars))
> summary(p)
  posterior.0      posterior.1
Min. :0.0001262   Min. :0.06317
1st Qu.:0.3547100 1st Qu.:0.25512
Median :0.4632872 Median :0.53671
Mean   :0.5065252 Mean  :0.49347
3rd Qu.:0.7448786 3rd Qu.:0.64529
Max.  :0.9368264 Max.  :0.99987
>
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_attitude_naive.txt')
>
> # calculation of the predicted probability values
>
> m_data = read.table('cyberbullying_attitude_naive.txt',header=T)
> #attach(m_data)
> mean(m_data$posterior.0)
[1] 0.5065251
> mean(m_data$posterior.1)
[1] 0.4934748
> |

```

Analysis: In the Naïve Bayes classification model, the dependent variables' mean prediction probability for negative was 50.65%, and the mean prediction probability for positive was 49.35%.

Cyber bullying Type Prediction Model

To develop a Naïve Bayes classification model that predicts the cyber bullying types (Perpetrator, Victim, Bystander, Complex), a total of 59,672 data [Perpetrator: 7,285 (12.2%), Victim: 42,237 (70.8%),

Bystander (9.4%), Complex (7.7%)], which mention the cyber bullying type and have the frequency of GST factors, were used as training data.

```
> tdata = read.table('cyberbullying_type_N.txt',header=T)
  - Assign the training data file to the tdata object.
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_type.txt',header=T,sep=",")
  - Assign the dependent variable to the output object as a delimiter (,).
> p_output=read.table('p_output_type.txt',header=T,sep=",")
  - Assign the Bayes model of predictive value to the output object as a
    delimiter (,).
> attach(tdata)
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> train_data.lda=naiveBayes(form,data=tdata)
  - Create a model function by executing the Naive Bayes Classification
    model with tdata dataset.
> p=predict(train_data.lda, tdata, type='raw')
  - Create a prediction by the type by executing a prediction model with
    tdata set.
> dimnames(p)=list(NULL,c(p_output_vars))
  - Assign the probability value of the predicted dependent variable to
    the variables p_Perpetrator (perpetrator prediction probability),
    p_Victim (victim prediction probability), p_Bystander (bystander
    prediction probability), and p_Complex (complex prediction
    probability).
> summary(p)
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_type_naive.txt')
> mydata1=read.table('cyberbullying_type_naive.txt',header=T)
> attach(mydata1)
> mean(mydata1$p_Perpetrator)
> mean(mydata1$p_Victim)
> mean(mydata1$p_Bystander)
> mean(mydata1$p_Complex)
```



```

R Console
> install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
> library(e1071)
> tdata = read.table('cyberbullying_type_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning message:
In read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
> p_output=read.table('p_output_type.txt',header=T,sep=",")
Warning message:
In read.table("p_output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_type.txt'
> #attach(tdata)
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')) )
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> train_data.lda=naiveBayes(form,data=tdata)
> p=predict(train_data.lda, tdata, type='raw')
>
> dimnames(p)=list(NULL,c(p_output_vars))
> #summary(p)
>
> pred_obs = cbind(tdata, p)
>
> write.matrix(pred_obs, 'cyberbullying_type_naive.txt')
>
> mydata1=read.table('cyberbullying_type_naive.txt',header=T)
> #attach(mydata1)
> mean(mydata1$p_Perpetrator)
[1] 0.2009258
> mean(mydata1$p_Victim)
[1] 0.6209646
> mean(mydata1$p_Bystander)
[1] 0.08342786
> mean(mydata1$p_Complex)
[1] 0.09468183
> |

```

Analysis: In the Naïve Bayes classification model, the dependent variables' mean prediction probability was 20.09% for perpetrator, 62.09% for victim, 8.34% for bystander, and 9.47% for complex.

Logistic Regression Model

The logistic regression model is a nonlinear regression model that has quantitative variables as its independent variables and qualitative variables as its dependent variables. Normally, a regression model's goodness-of-fit test uses the least-squares method, which minimizes the residual sum of

squares. However, the logistic regression model uses the maximum likelihood method, which maximizes the likelihood of an event occurring.

The logistic regression model tests the effect of the independent variable (or covariate) on the dependent variable via the odds ratio, which is the probability of success. Therefore, the logit model of the odds ratio of the probability ratio for predicting a binary or dichotomous logistic regression model appears as ($\ln \frac{P(Y=1|X)}{P(Y=0|X)} = \beta_0 + \beta_1 X$). The categories of the dependent variables are (0, 1). Here, the regression coefficient estimates changes occurring in the odds ratio. The odds ratio is calculated by replacing the regression coefficient with inverse log.

Multinomial or polychotomous logistic regression has continuous values as the independent variables. It also has multinomial categories, which have three or more categories for the dependent variables.

Cyber bullying Risk (Negative, Positive) Prediction Model

The binary logistic regression model for predicting emotions (Positive, Negative) about cyber bullying is as follows.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> p_output=read.table('p_output_bayes.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> i_logistic=glm(form, family=binomial,data=tdata)
  - Create a model function (classifier) by executing the binary logistics
    regression model with tdata dataset.
> p=predict(i_logistic,tdata,type='response')
  - Create risk prediction groups by executing model predictions with the
    tdata dataset.
> mean(p)
  - Dispaly the positive predictive probability for cyber bullying on the
    screen.
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_attitude_logistic.txt')
```



```

> #2 logistic regression modeling(attitude)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
> p_output=read.table('p_output_bayes.txt',header=T,sep=",")
Warning message:
In read.table("p_output_bayes.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_bayes.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
   Attachment + Passion + Offender_psychology + Delinquency
>
> i_logistic=glm(form, family=binomial,data=tdata)
> #summary(i_logistic)
>
> p=predict(i_logistic,tdata,type='response')
> mean(p)
[1] 0.5467768
>
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_attitude_logistic.txt')
> |
<
```

Analysis: In the binary logistic regression model, the dependent variables' mean prediction probability was 54.68% for positive and 45.32% for negative.

Cyber bullying Type Prediction Model

The multinomial logistic regression model for predicting types for cyber bullying (Perpetrator, Victim, Bystander, Complex) is as follows.

```

> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("nnet")
  - Install the package 'nnet' for polynomial logistic regression analysis.
> library(nnet)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_type_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_type.txt',header=T,sep=",")
> p_output=read.table('p_output_type.txt',header=T,sep=",")
```

```
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
> form
> i_logistic=multinom(form, data=tdata)
> p=predict(i_logistic,tdata,type='probs')
> dimnames(p)=list(NULL,c(p_output_vars))
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_type_logistic.txt')
> m_data = read.table('cyberbullying_type_logistic.txt',header=T)
> mean(m_data$p_Perpetrator)
> mean(m_data$p_Victim)
> mean(m_data$p_Bystander)
> mean(m_data$p_Complex)
```



The screenshot shows the R Console window with the following text:

```
R Console

> output=read.table('output_type.txt',header=T,sep=",")
Warning message:
In read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
> p_output=read.table('p_output_type.txt',header=T,sep=",")
Warning message:
In read.table("p_output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_type.txt'
> # logistic modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+   paste(input_vars, collapse = '+'))))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> i_logistic=multinom(form, data=tdata)
# weights:  40 (27 variable)
initial  value 82722.957117
iter  10  value 73392.284410
iter  20  value 65307.080038
iter  30  value 55824.495948
iter  40  value 53154.419871
final  value 53154.356454
converged
>
> p=predict(i_logistic,tdata,type='probs')
>
> dimnames(p)=list(NULL,c(p_output_vars))
> #summary(p)
>
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_type_logistic.txt')
>
> m_data = read.table('cyberbullying_type_logistic.txt',header=T)
> #attach(m_data)
> mean(m_data$p_Perpetrator)
[1] 0.1220865
> mean(m_data$p_Victim)
[1] 0.7078182
> mean(m_data$p_Bystander)
[1] 0.09352824
> mean(m_data$p_Complex)
[1] 0.07656706
```

Analysis: In the multinomial logistic regression model, the dependent variable's mean prediction probability was 12.21% for perpetrator, 70.78% for victim, 9.35% for bystander, and 7.66% for complex.

Random Forest Model

The random forest model was proposed by Breiman (2001). It is an ensemble technique for machine learning in which several prediction models are created for a given data set and combined to create one final prediction model. It has the advantages of excellent classification accuracy, not being sensitive to outliers, and making fast calculations (Jin & Oh, 2013). The first ensemble algorithm was the bagging (bootstrap aggregating) algorithm proposed by Breiman (1996).

Bagging is a method for improving the predictive power by removing unstable learning methods, a disadvantage of the decision tree; These simultaneously complicate the prediction model analysis and reduce the predictive power because their ultimate decision tree is different if the first separation variable is changed. The bagging method creates several bootstrap data to create the prediction model. Then, it combines them to create the ultimate model.

Random forest creates bootstrap samples, which use n data sets from the training data, and randomly selects some of the input variables to create decision trees. These are linearly combined to create the final model function (classifier). Random forest provides an importance index for the variables. The importance index of a particular variable shows the extent to which the prediction error is reduced when the variable is included, in a case where it is not already included.

When terminal nodes are present, random forest determines the classification of dependent variables through the majority of terminal nodes. In random forest, the Mean Decrease Accuracy (%IncMSE) is the most robust and informative measure, and it shows the method's accuracy. Mean Decrease Gini (IncNodePurity) relates to the loss function by which the best splits are chosen, and it shows the method's importance.

Cyber bullying Risk (Negative, Positive) Prediction Model

The random forest model for predicting emotions (Positive, Negative) about cyber bullying is as follows.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("randomForest")
> library(randomForest)
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> tdata.rf = randomForest(form, data=tdata , forest=FALSE, importance=
  TRUE)
> p=predict(tdata.rf,tdata)
> mean(p)
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_attitude_randomforest.txt')
> varImpPlot(tdata.rf, main='Random forest importance plot')
  - random forest: Display the important diagram for the prediction
  model on the screen.
```



```

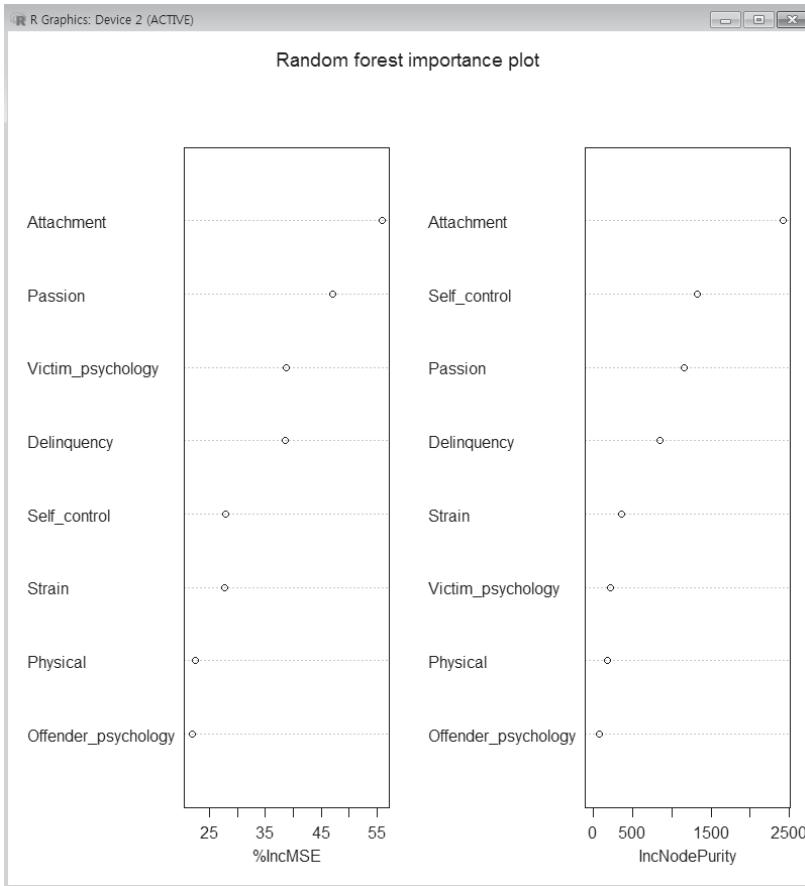
R Console
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("randomForest")
trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.5/randomForest.zip'
Content type 'application/zip' length 248452 bytes (242 KB)
downloaded 242 KB

package 'randomForest' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\SAMSUNG\AppData\Local\Temp\Rtmp042D6c\downloaded_packages
> library(randomForest)
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
>
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> # random forests modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> tdata.rf = randomForest(form, data=tdata ,forest=FALSE,importance=TRUE)
Warning message:
In randomForest.default(m, y, ...):
  The response has five or fewer unique values. Are you sure you want to do a
> p=predict(tdata.rf,tdata)
> mean(p)
[1] 0.5467073
>
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_attitude_randomforest.txt')
>
> varImpPlot(tdata.rf, main='Random forest importance plot')
> |

```

Analysis: In the random forest model, the dependent variable's mean prediction probability was 54.68% for positive and 45.32% for negative.



Analysis: In the random forest's importance plot, the GST factor that had the most effect on the cyber bullying emotion (positive, negative) was Attachment, followed by Self control, Passion, Delinquency, Strain, and so on.

Cyber bullying Type Prediction Model

The random forest model for predicting types (Perpetrator, Victim, Bystander, Complex) for cyber bullying is as follows.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("randomForest")
> library(randomForest)
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_type.txt',header=T,sep=",")
> p_output=read.table('p_output_type_random.txt',header=T,sep=",")
- Assign the predictions of the random forest model as p_output
  (p_Bystander, p_Complex_psychology, p_Perpetrator, and p_Victim)
  to the object as delimiters (,).
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
> form
> tdata.rf = randomForest(form, data=tdata ,forest=FALSE,importance=
  TRUE)
> p=predict(tdata.rf,tdata,type='prob')
> dimnames(p)=list(NULL,c(p_output_vars))
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_type_random.txt')
> mydata1=read.table('cyberbullying_type_random.txt',header=T)
> attach(mydata1)
> mean(mydata1$p_Perpetrator)
> mean(mydata1$p_Victim)
> mean(mydata1$p_Bystander)
> mean(mydata1$p_Complex)
> varImpPlot(tdata.rf, main='Random forest importance plot')
```



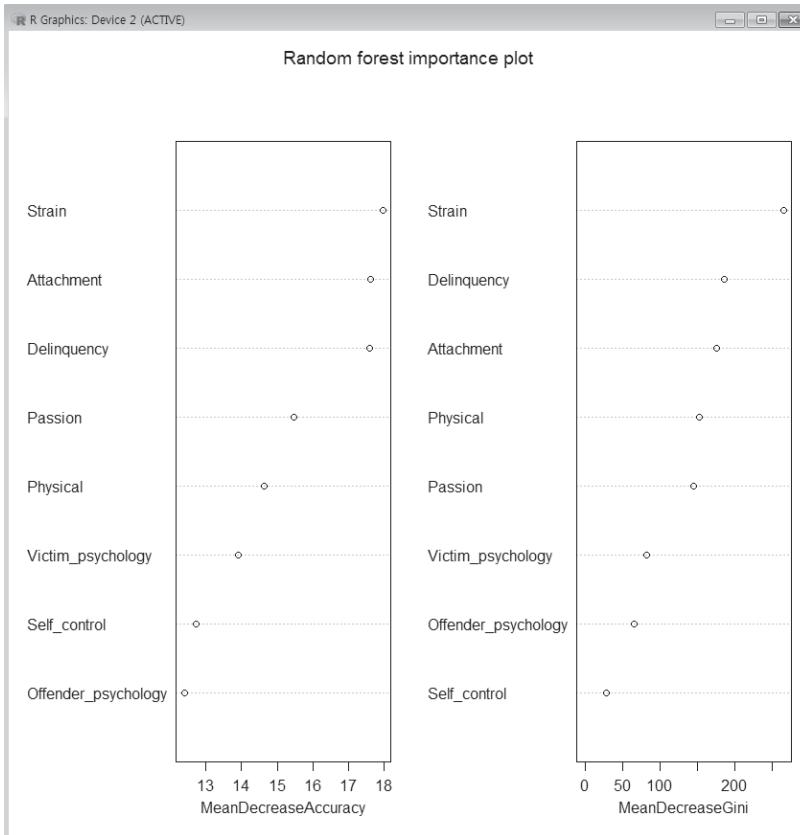
```

R Console

> setwd("c:/cyberbullying_2017")
> install.packages("randomForest")
Warning: package 'randomForest' is in use and will not be installed
> library(randomForest)
>
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning message:
In read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
> p_output=read.table('p_output_type_random.txt',header=T,sep=",")
Warning message:
In read.table("p_output_type_random.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_type_random.txt'
>
> # random forests modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
> tdata_rf = randomForest(form, data=tdata ,forest=FALSE,importance=TRUE)
> p=predict(tdata_rf,tdata,type='prob')
> dimnames(p)=list(NULL,c(p_output_vars))
>
> pred_obs = cbind(tdata, p)
>
> write.matrix(pred_obs,'cyberbullying_type_random.txt')
> mydata1=read.table('cyberbullying_type_random.txt',header=T)
> #attach(mydata1)
>
> mean(mydata1$p_Perpetrator)
[1] 0.01983902
> mean(mydata1$p_Victim)
[1] 0.9795809
> mean(mydata1$p_Bystander)
[1] 9.334361e-05
> mean(mydata1$p_Complex)
[1] 0.0004866939
> |

```

Analysis: In the random forest model, the dependent variable's mean prediction probability was 1.98% for perpetrator, 97.96% for victim, 0.00093% for bystander, and 0.047% for complex.



Analysis: In the random forest's importance plot, the GST factor that had the most effect on the cyber bullying type was Strain, followed by Delinquency, Attachment, Physical, Passion, Victim psychology, Offender psychology and Self control.

- If you see the error message "Error: can not allocate vector of size 3.3 Gb" when executing the random forest algorithm, you can run virtual memory with the following statement.

```
> memory.size(22000): Allocate 2.2 Gb of virtual memory.
```

Decision Tree Model

The decision tree model classifies and predicts by diagramming a tree structure according to the decision rules. It is a data-mining technique that combines discriminant analysis and regression analysis. Data mining is the science of extracting useful information from large data sets or databases (Hand et al., 2001: p. 2).

Decision tree models are suitable for segmentation, classification, clustering, and forecasting. The advantages of the decision tree model are that it is easy to determine which predictor variable is more important for describing a target variable, and that it is easy to determine what effect two or more combined variables will have on a target variable (U.S. EPS, 2003). In decision tree analysis, a variety of algorithms have been proposed for quickly and accurately forming decision trees, through a combination of separation criteria, stopping rules, and pruning methods.

Representative decision tree algorithms include CHAID, CRT, and QUEST.

Cyber bullying Risk (Negative, Positive) Prediction Model

The decision tree model for predicting emotions (Negative, Positive) about cyber bullying is as follows.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('party')
> library(party)
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> p_output=read.table('p_output_decision.txt',header=T,sep=",")
    - Assign the prediction value (p_attitude) of the decision model to the
      p_output object as a delimiter (,).
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
> paste(input_vars, collapse = '+'))))
> form
> i_ctree=ctree(form, tdata)
> p=predict(i_ctree, tdata)
```

```
> dimnames(p)=list(NULL,c(p_output_vars))
- Assign the probability value of the predicted dependent variable to
the p_Attitude variable.
> mean(p)
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_attitude_decision.txt')
> mydata1=read.table('cyberbullying_attitude_decision.txt',header=T)
> attach(mydata1)
> mean(mydata1$p_Attitude)
```



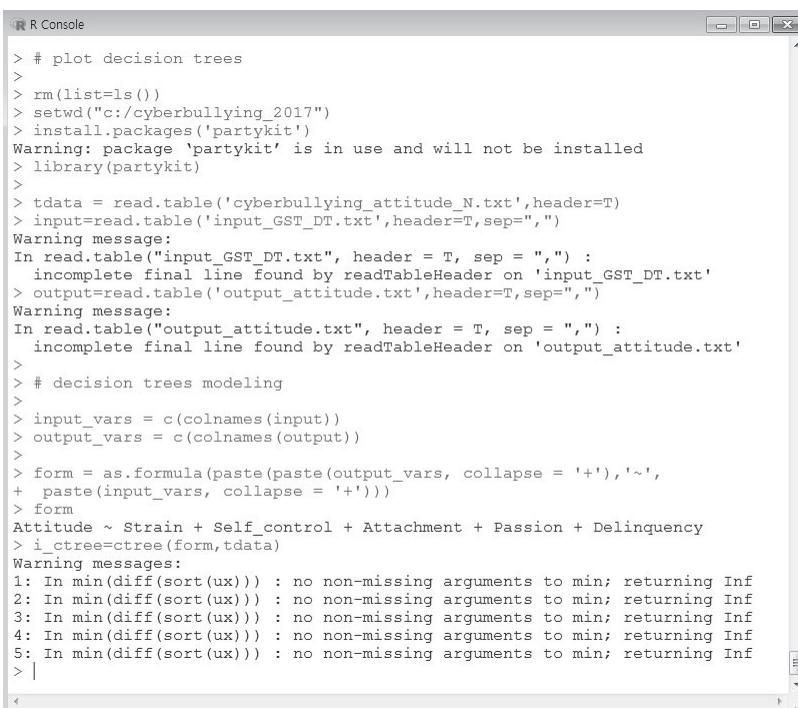
The screenshot shows an R console window with the title 'R Console'. The window contains the R code from the previous block, which is used to build a decision tree model. The code includes several warning messages related to incomplete final lines found by `readTableHeader` and `readTable`. The output shows the formula for the model, the creation of a decision tree object (`i_ctree`), and the prediction of probabilities for the 'p_Attitude' variable. The mean probability for the positive class is printed as 0.5467768.

```
> #4 decision trees modeling(attitude)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('party')
Warning: package 'party' is in use and will not be installed
> library(party)
>
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
> p_output=read.table('p_output_decision.txt',header=T,sep=",")
Warning message:
In read.table("p_output_decision.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_decision.txt'
>
> # decision trees modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> i_ctree=ctree(form,tdata)
> p=predict(i_ctree,tdata)
> dimnames(p)=list(NULL,c(p_output_vars))
> mean(p)
[1] 0.5467768
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_attitude_decision.txt')
> mydata1=read.table('cyberbullying_attitude_decision.txt',header=T)
> #attach(mydata1)
> mean(mydata1$p_Attitude)
[1] 0.5467768
> |
```

Analysis: In the decision tree model, the dependent variable's mean prediction probability was 54.68% for positive and 45.32% for negative.

- A decision tree analysis was performed on five factors (Strain, Self_control, Attachment, Passion, and Delinquency) that were shown to be important factors in the random forest model's importance analysis. A graph of the results is shown below.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('partykit')
> library(partykit)
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST_DT.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
> i_ctree=ctree(form,tdata)
```



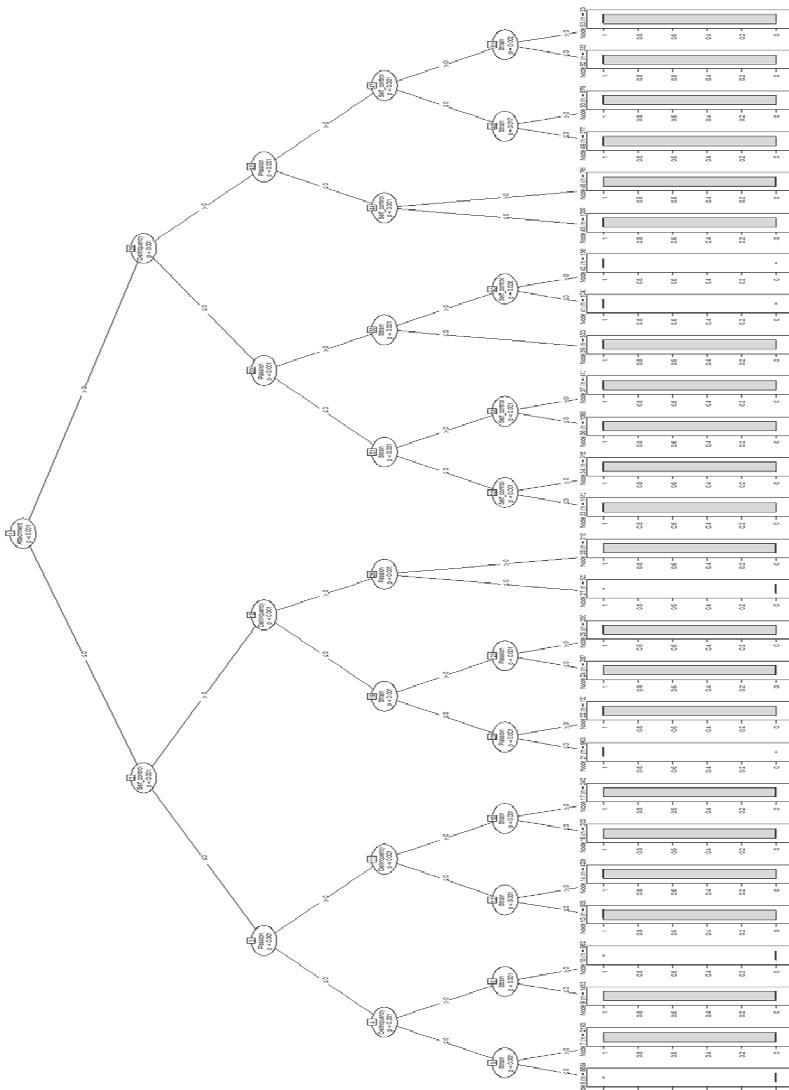
The screenshot shows the R Console window with the following content:

```
R Console
```

```
> # plot decision trees
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('partykit')
Warning: package 'partykit' is in use and will not be installed
> library(partykit)
>
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST_DT.txt',header=T,sep=",")
Warning message:
In read.table("input_GST_DT.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST_DT.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> # decision trees modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Self_control + Attachment + Passion + Delinquency
> i_ctree=ctree(form,tdata)
Warning messages:
1: In min(diff(sort(ux))) : no non-missing arguments to min; returning Inf
2: In min(diff(sort(ux))) : no non-missing arguments to min; returning Inf
3: In min(diff(sort(ux))) : no non-missing arguments to min; returning Inf
4: In min(diff(sort(ux))) : no non-missing arguments to min; returning Inf
5: In min(diff(sort(ux))) : no non-missing arguments to min; returning Inf
> |
```

```
> print(i_ctree)
> plot(i_ctree, gp=gpar(fontsize=6))
```

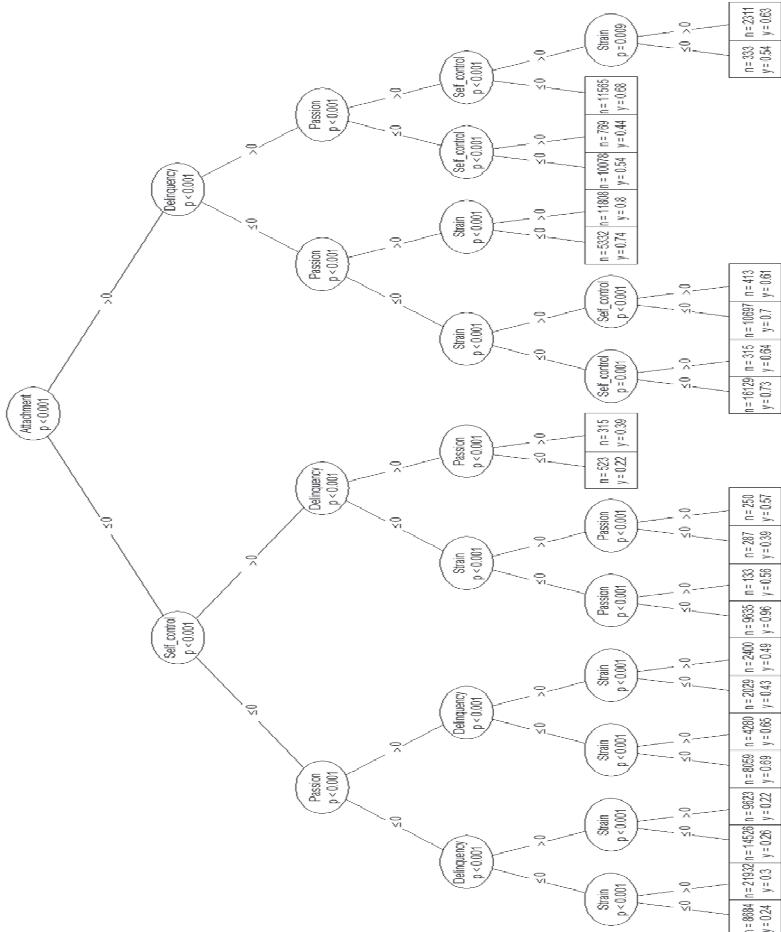
```
R Console
Fitted party:
[1] root
| [2] Attachment <= 0
| | [3] Self_control <= 0
| | | [4] Passion <= 0
| | | | [5] Delinquency <= 0
| | | | | [6] Strain <= 0: 0.245 (n = 8684, err = 1604.5)
| | | | | [7] Strain > 0: 0.302 (n = 21932, err = 4626.2)
| | | | [8] Delinquency > 0
| | | | | [9] Strain <= 0: 0.264 (n = 14526, err = 2822.5)
| | | | | [10] Strain > 0: 0.223 (n = 9623, err = 1666.9)
| | | [11] Passion > 0
| | | | [12] Delinquency <= 0
| | | | | [13] Strain <= 0: 0.691 (n = 8059, err = 1721.4)
| | | | | [14] Strain > 0: 0.654 (n = 4280, err = 968.8)
| | | | [15] Delinquency > 0
| | | | | [16] Strain <= 0: 0.432 (n = 2029, err = 497.8)
| | | | | [17] Strain > 0: 0.488 (n = 2400, err = 599.7)
| | | [18] Self_control > 0
| | | | [19] Delinquency <= 0
| | | | | [20] Strain <= 0
| | | | | | [21] Passion <= 0: 0.956 (n = 9635, err = 403.5)
| | | | | | [22] Passion > 0: 0.564 (n = 133, err = 32.7)
| | | | | [23] Strain > 0
| | | | | | [24] Passion <= 0: 0.390 (n = 287, err = 68.3)
| | | | | | [25] Passion > 0: 0.568 (n = 250, err = 61.3)
| | | | | [26] Delinquency > 0
| | | | | | [27] Passion <= 0: 0.222 (n = 523, err = 90.3)
| | | | | | [28] Passion > 0: 0.390 (n = 315, err = 75.0)
| [29] Attachment > 0
| | [30] Delinquency <= 0
| | | [31] Passion <= 0
| | | | [32] Strain <= 0
| | | | | [33] Self_control <= 0: 0.731 (n = 16129, err = 3174.5)
| | | | | [34] Self_control > 0: 0.638 (n = 315, err = 72.7)
| | | | [35] Strain > 0
| | | | | [36] Self_control <= 0: 0.698 (n = 10697, err = 2256.3)
| | | | | [37] Self_control > 0: 0.608 (n = 413, err = 98.5)
| | | [38] Passion > 0
| | | | [39] Strain <= 0: 0.742 (n = 5332, err = 1020.4)
| | | | [40] Strain > 0
| | | | | [41] Self_control <= 0: 0.808 (n = 10439, err = 1621.7)
| | | | | [42] Self_control > 0: 0.782 (n = 1369, err = 233.1)
| | [43] Delinquency > 0
| | | [44] Passion <= 0
| | | | [45] Self_control <= 0: 0.537 (n = 10078, err = 2505.8)
| | | | [46] Self_control > 0: 0.440 (n = 769, err = 189.4)
| | | [47] Passion > 0
| | | | [48] Self_control <= 0
```



Analysis: The root node at the very top of the tree structure shows the frequency of the dependent variables without the independent variables inserted. The variables located just below the root node are the variables with the most effect (highest association) on the dependent variables. This shows that Attachment has the most effect on cyberbullying risk emotion.

■ CTREE Node %[y=(negative, positive)] Output

```
> install.packages('party')
> library(party)
> i_ctree=ctree(form,tdata)
> plot(i_ctree, type="simple", inner_panel=node_inner (i_ctree,abbreviate
 = FALSE, pval = TRUE, id = FALSE), terminal_panel=
node_terminal(i_ctree, abbreviate = FALSE, digits = 2, fill = c("white"),
id = FALSE))
> nodes(i_ctree, 2)
```



Cyber bullying Type Prediction Model

The decision tree model for predicting types (Perpetrator, Victim, Bystander, Complex) for cyber bullying is as follows.

```
> install.packages('party')
> library(party)
> install.packages('MASS')
> library(MASS)
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_type.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> i_ctree=ctree(form,tdata)
> Bystander1=sapply(predict(i_ctree,tdata,type='prob'),'[',1)


- sapply is wrapper class to lapply with difference being it returns vector or matrix instead of list object. lapply function is applied for operations on list objects and returns a list object of same length of original set.
- Calculate the prediction probability by the model prediction with the tdata set, and then assign the first probability value to Bystander1


> Complex1=sapply(predict(i_ctree,tdata,type='prob'),'[',2)


- Calculate the prediction probability by the model prediction with tdata set, and then assign the second probability value to Complex1.


> Perpetrator1=sapply(predict(i_ctree,tdata,type='prob'),'[',3)


- Calculate the prediction probability by the model prediction with tdata set, and then the third probability value to Perpetrator1.


> Victim1=sapply(predict(i_ctree,tdata,type='prob'),'[',4)


- Calculate the prediction probability by the model prediction with tdata set, and then the third probability value to Victiom1.


> mydata=cbind(tdata,Perpetrator1,Victim1,Bystander1,Complex1)


- Append the Perpetrator1, Victim1, Bystander1, and complex1 variables to the tdata set and then assign them to the 'mydata' object.


> write.matrix(mydata,'decision_trees_cyberbullying_type.txt')


- Save the mydata object as 'decision_trees_cyberbullying_type.txt' file.


> mydata1=read.table('decision_trees_cyberbullying_type.txt',header=T)
```

- Assign the ‘decision_trees_cyberbullying_type.txt’ file to the ‘mydata1’ object.

```
> attach(mydata1)
> mean(mydata1$Perpetrator1)
> mean(mydata1$Victim1)
> mean(mydata1$Bystander1)
> mean(mydata1$Complex1)
```

The screenshot shows the R console window with the following content:

```

R Console

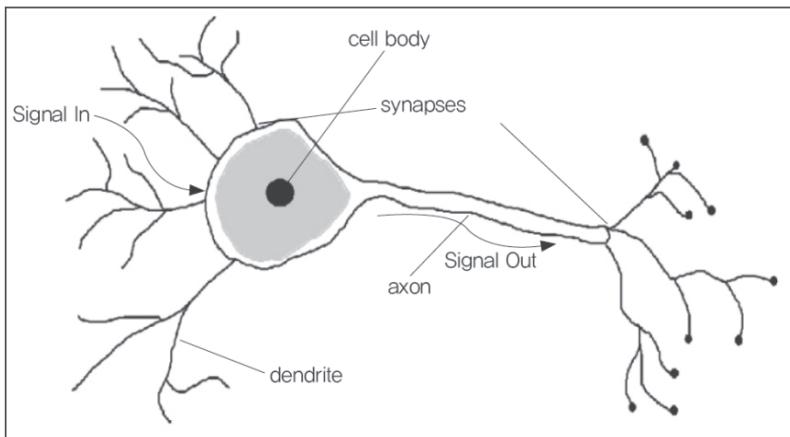
> library(MASS)
> install.packages('party')
Warning: package 'party' is in use and will not be installed
> library(party)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning message:
In read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
>
> # decision trees modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> i_ctree=ctree(form,tdata)
>
> # allocation of the predicted probability values
>
> Bystander1=sapply(predict(i_ctree,tdata,type='prob'), '[[,1]
> Complex1=sapply(predict(i_ctree,tdata,type='prob'), '[[,2]
> Perpetrator1=sapply(predict(i_ctree,tdata,type='prob'), '[[,3]
> Victim1=sapply(predict(i_ctree,tdata,type='prob'), '[[,4]
>
> mydata=cbind(tdata,Perpetrator1,Victim1,Bystander1,Complex1)
> write.matrix(mydata,'decision_trees_cyberbullying_type.txt')
> mydata1=read.table('decision_trees_cyberbullying_type.txt',header=T)
> #attach(mydata1)
> mean(mydata1$Perpetrator1)
[1] 0.1220841
> mean(mydata1$Victim1)
[1] 0.7078194
> mean(mydata1$Bystander1)
[1] 0.09352795
> mean(mydata1$Complex1)
[1] 0.07656857
> |
```

Analysis: In the decision tree model, the dependent variable’s prediction probability was 12.21% for perpetrator, 70.78% for victim, 9.35% for bystander, and 7.65% for complex.

Neural Network Model

Artificial neural networks are a type of machine learning model that uses the operation of biological neural networks, e.g., the human nervous system, as its basic concept. The classification process imitates how the human brain makes decisions.

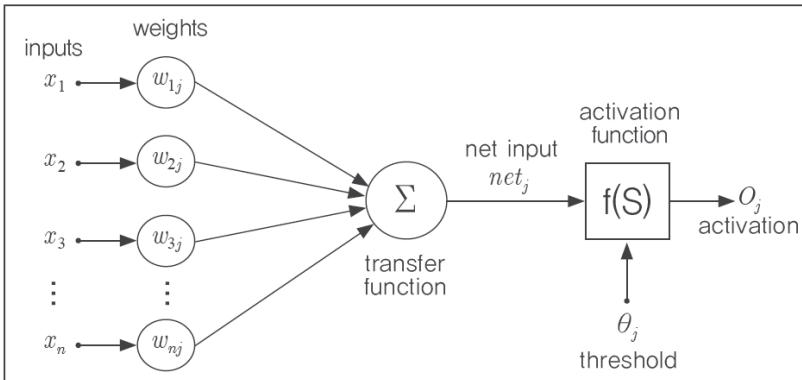
The biological (Human) neural network is composed of 25 billion neurons. Each neuron is composed of one cell body, one axon, which protrudes from the cell body, and several dendrites. Information is exchanged between neurons through connective parts called synapses. Synapses do not transmit the neurons' signal unconditionally; they transmit it only if the signal strength is above a set value (threshold). The cell body accumulates the signal, which enters from the dendrites, and when it reaches the threshold value, the output signal is transmitted to the axon. From there, it is transmitted to surrounding neurons via the synapses at the end of the axon (Fig. 5).



Reference: <https://cogsci.stackexchange.com/questions/7880/what-is-the-difference-between-biological-and-artificial-neural-networks>

Fig. 5. Biological Neural Network

Neural networks are supervised learning methods that imitate the human brain structure. Multiple neurons are connected to each other to predict the optimal output value for a given input value. In neural networks, signals are received from the training data, as occurs in neurons, the basic unit of the brain, and if the input value reaches a certain threshold, output occurs (Fig. 6).



Reference: https://commons.wikimedia.org/wiki/File:Artificial_neural_network.png
Fig.6. Artificial Neural Network

Minsky and Papert (1969) introduced a hidden layer to a single-layer neural network (perceptron), which can only solve linear problems, and showed that it was possible to classify using a generalized nonlinear function. Rumelhart et al. (1986) performed back propagation on the output layer's error to develop a back-propagation algorithm that can train the hidden layer. Deep learning is the creation of deep neural networks: Multilayer neural networks that have a large number of hidden layers between the input and the output layers. Fig. 7 shows an example for predicting cyber bullying risk. A multilayer neural network is composed of an input layer, made of input nodes, hidden layers, which are collections of intermediate nodes composed of input-layer nodes, and an output layer, which is composed of hidden-layer nodes.

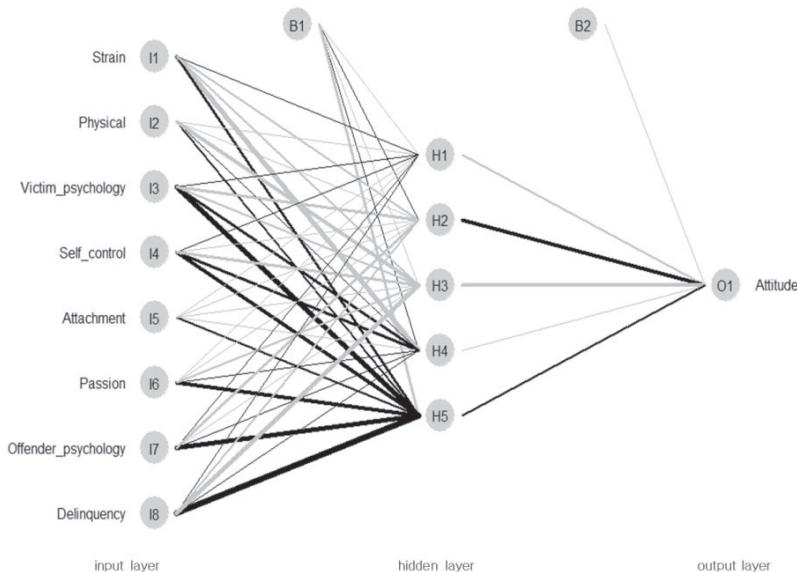


Fig. 7. Multilayer neural network for cyber bullying risk-prediction

The neural network's output node (O1) is calculated by the population's predicted value (\hat{y}) which is obtained by linear combining of the input nodes and the hidden nodes. A linear combination is performed on the weighted coefficients between the input nodes and the hidden nodes to perform the computation. The combination or transfer function performs the linear combination in the neural network, and the activation function is used to examine the range of the combination function's values. The activation function creates the output value (signal) when the input value (signal) crosses a certain threshold; it then converts the combination function's values into attached-range-values (-1, 0, 1). In short, a neural network receives input values and creates a combination function; then, the activation function generates output values. One activation function, the sigmoid function ($y = \frac{1}{1+e^{-x}}$) of the S character form, transforms the input values to values between 0 and 1. If the input variables' values are very large or small, the sigmoid function has almost no effect on the output variables' values; hence, the sigmoid function is often used in neural-network training algorithms. The formula for calculating the predicted values (\hat{y}) of the multilayer neural network in Fig. 7 is shown below.

$$O_{H1} = f_{H1} \left(\sum_{i=1}^8 w_{IiH1} I_i + w_{B1H1} \right) \quad (1)$$

$$O_{H2} = f_{H2} \left(\sum_{i=1}^8 w_{IiH2} I_i + w_{B1H2} \right) \quad (2)$$

$$O_{H3} = f_{H3} \left(\sum_{i=1}^8 w_{IiH3} I_i + w_{B1H3} \right) \quad (3)$$

$$O_{H4} = f_{H4} \left(\sum_{i=1}^8 w_{IiH4} I_i + w_{B1H4} \right) \quad (4)$$

$$O_{H5} = f_{H5} \left(\sum_{i=1}^8 w_{IiH5} I_i + w_{B1H5} \right) \quad (5)$$

$$\hat{y} = f_{O1} (w_{H1O1} H1 + w_{H2O1} H2 + w_{H3O1} H3 + w_{H4O1} H4 + w_{H5O1} H5 + w_{B2O1}) \quad (6)$$

Here, $I1 \sim I8$ are the input nodes, $H1 \sim H5$ are the hidden nodes, and $O1$ is the output node. $B1$ and $B2$ are the biases of the linear model. $w_{IiH1} \sim w_{IiH5}$ are the weight coefficients of the connections between the input nodes and the hidden nodes. ' w_{B1H1} , w_{B1H2} , w_{B1H3} , w_{B1H4} , w_{B1H5} , w_{B2O1} ' are the bias terms. $f_{H1} \sim f_{H5}$ are the hidden nodes' activation functions. f_{O1} is the output node's activation function. $O_{H1} \sim O_{H5}$ are the output values calculated at hidden nodes $H1 \sim H5$. \hat{y} is the value of y estimated by the nonlinear combination function.

Multilayer neural networks perform an approximation using combination functions and activation functions, so the analysis process is not seen. So this is called black box analysis.

The following items must be considered when designing a multilayer neural network model. First, the range of the input variable values must be determined. For data to be suitable for a neural network model, categorical variables must have values that are above a fixed frequency for all categories. Continuous variables must be converted into categorical variables, or their values must be converted to values 0 and 1.

Second, the number of hidden layers and hidden nodes must be set properly. If the number of hidden layers and hidden nodes is too high, the weight coefficients become too numerous, which creates the possibility of overfitting. Therefore, in many cases when using neural network models, the number of hidden layers is set to one, and the number of hidden nodes is set so that it is adequate. As the number of hidden nodes is progressively reduced by one and the classification accuracy is raised, a model with a

small number of hidden nodes is chosen.

Cyber bullying Risk (Negative, Positive) Prediction Model

The neural network model for predicting emotions (Negative, Positive) about cyber bullying is as follows. Neural network model analysis in R uses the “nnet” and “neuralnet” packages.

■ Using the ‘nnet’ package

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("nnet")
> library(nnet)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> tr.nnet = nnet(form, data=tdata, size=5)
  - Create a model function (classifier) by executing a neural network
    model with five hidden layer in the tdata dataset.
> p=predict(tr.nnet, tdata, type='raw')
  - Generate the risk prediction groups by executing predictions model
    with the tdata set.
> mean(p)
> mydata=cbind(tdata, p)
> write.matrix(mydata,'cyberbullying_attitude_neural.txt')
> mydata1=read.table('cyberbullying_attitude_neural.txt',header=T)
> attach(mydata1)
> mean(p)
```



The screenshot shows the R Console window with the following R script:

```

> library(MASS)
>
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> tr.nnet = nnet(form, data=tdata, size=5)
# weights:  51
initial value 37341.031923
iter  10 value 30019.913637
iter  20 value 29743.075347
iter  30 value 29611.094801
iter  40 value 29536.862571
iter  50 value 29360.564462
iter  60 value 29194.110636
iter  70 value 29146.349821
iter  80 value 29130.876343
iter  90 value 29124.479102
iter 100 value 29123.241189
final value 29123.241189
stopped after 100 iterations
> p=predict(tr.nnet, tdata, type='raw')
> mean(p)
[1] 0.5469054
>
> mydata=cbind(tdata, p)
> write.matrix(mydata,'cyberbullying_attitude_neural.txt')
>
> mydata1=read.table('cyberbullying_attitude_neural.txt',header=T)
> #attach(mydata1)
> mean(p)
[1] 0.5469054
> |

```

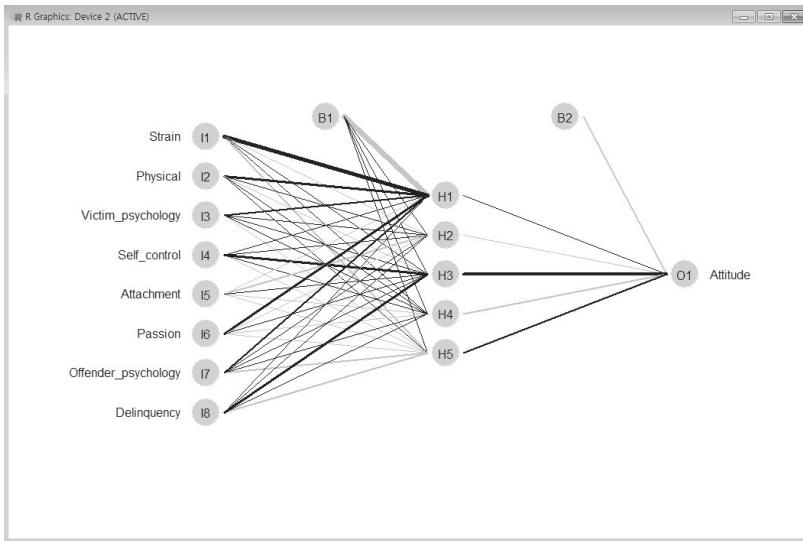
Analysis: In the neural network model using “nnet,” the dependent variable’s mean prediction probability was 54.67% for positive and 45.33% for negative.

```

> install.packages('NeuralNetTools')
  - Install the package (NeuralNetTools) that displays on the screen a
    picture of the neural network model analyzed ‘nnet’ package.
> library(NeuralNetTools)
> plotnet(tr.nnet)

```

- Display a picture of the neural network model analyzed by the ‘nnet’ package on the screen.



■ Using the ‘neuralnet’ package

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('neuralnet')
> library(neuralnet)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> p_output=read.table('p_output_attitude_neuralnet.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
```

```
> net = neuralnet(form, tdata, hidden=5, lifesign = "minimal",
linear.output = FALSE, threshold = 0.1) # hidden=c(5,3)
- Create a model function (classifier) by executing a neural network
model with five hidden layers in the tdata set.
- threshold a numeric value specifying the threshold for the partial
derivatives of the error function as stopping criteria.
- linear.output logical. If act.fct should not be applied to the output
neurons set linear output to TRUE, otherwise to FALSE.
- lifesign a string specifying how much the function will print during
the calculation of the neural network. 'none', 'minimal' or 'full'.
> summary(net)
> plot(net)
- black line: weight between the layer and the connection.
- blue line:- The bias term at each step.
> pred = net$net.result[[1]]
- Calculated the prediction probability value and then assign to the
pred variable.
- net.result [[1]] means the MSE(mean square error) as the predicted
value of the distance from the predicted value (real data).
> dimnames(pred)=list(NULL,c(p_output_vars))
- Assign the p_output_vars to pred matrix.
> summary(pred)
> pred_obs = cbind(tdata, pred)
- Add the predicted probability value (P) to the tdata set.
> write.matrix(pred_obs,'cyberbullying_attitude_neuralnet.txt')
> m_data = read.table('cyberbullying_attitude_neuralnet.txt',header=T)
> attach(m_data)
> mean(p_Attitude): calculation of average prediction probability.
```

```

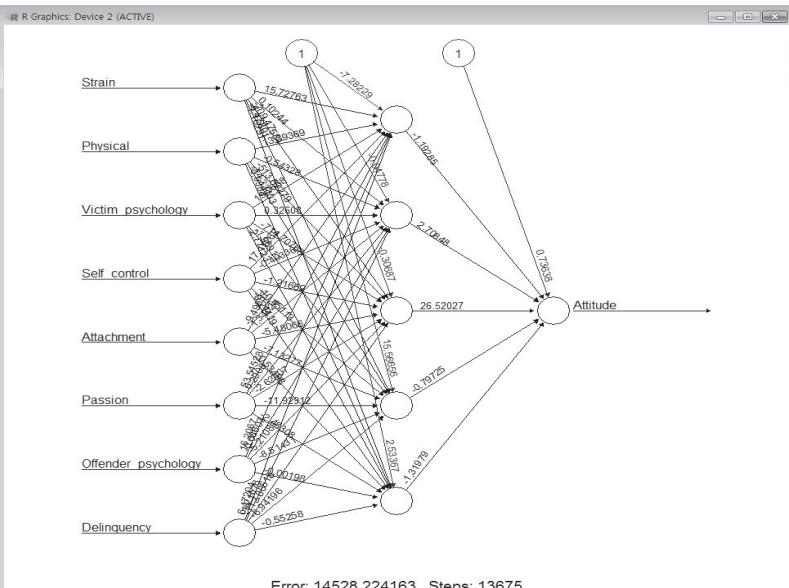
R Console

> # neural networks modeling neuralnet(attitude)
>
> rm(list=ls())
> setwd("~/cyberbullying_2017")
> install.packages('neuralnet')
Warning: package 'neuralnet' is in use and will not be installed
> library(neuralnet)
> install.packages('MASS')

There is a binary version available but the source version is later:
  binary source needs_compilation
MASS 7.3-49 7.3-50          TRUE

Binaries will be installed
Warning: package 'MASS' is in use and will not be installed
> library(MASS)
>
> tdata = read.table("cyberbullying_attitude_N.txt",header=T)
> input=read.table('input_GST.txt',header=T,sep="")
Warning message:
In read.table("input_GST.txt", header = T, sep = "") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table("output_attitude.txt",header=T,sep="")
Warning message:
In read.table("output_attitude.txt", header = T, sep = "") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
> p_output=read.table('p_output_attitude_neuralnet.txt',header=T,sep="")
Warning message:
In read.table("p_output_attitude_neuralnet.txt", header = T, sep = "") :
  incomplete final line found by readTableHeader on 'p_output_attitude_neuralnet.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
> net = neuralnet(form, tdata, hidden=5, lifesesign = "minimal",
+ linear.output = FALSE, threshold = 0.1)
hidden: 5  thresh: 0.1  rep: 1/1  steps: 13675  error: 14528.22416      time: 19.08 mins
> plot(net)
> |

```





```

R Console

> pred = net$net.result[[1]]
> dimnames(pred)=list(NULL,c(p_output_vars))
> summary(pred)
  p_Attitude
Min.   :0.1419057
1st Qu.:0.2860442
Median :0.6049433
Mean   :0.5467072
3rd Qu.:0.7281635
Max.   :0.9660731
>
> pred_obs = cbind(tdata, pred)
> write.matrix(pred_obs, 'cyberbullying_attitude_neuralnet.txt')
>
> # calculation of the predicted probability values
>
> m_data = read.table('cyberbullying_attitude_neuralnet.txt',header=T)
> #attach(m_data)
> mean(m_data$p_Attitude)
[1] 0.5467071685
> |

```

Analysis: In the neural network model using “neuralnet,” the dependent variable’s mean prediction probability was 54.70% for positive and 45.30% for negative.

■ Create ROC curve

```

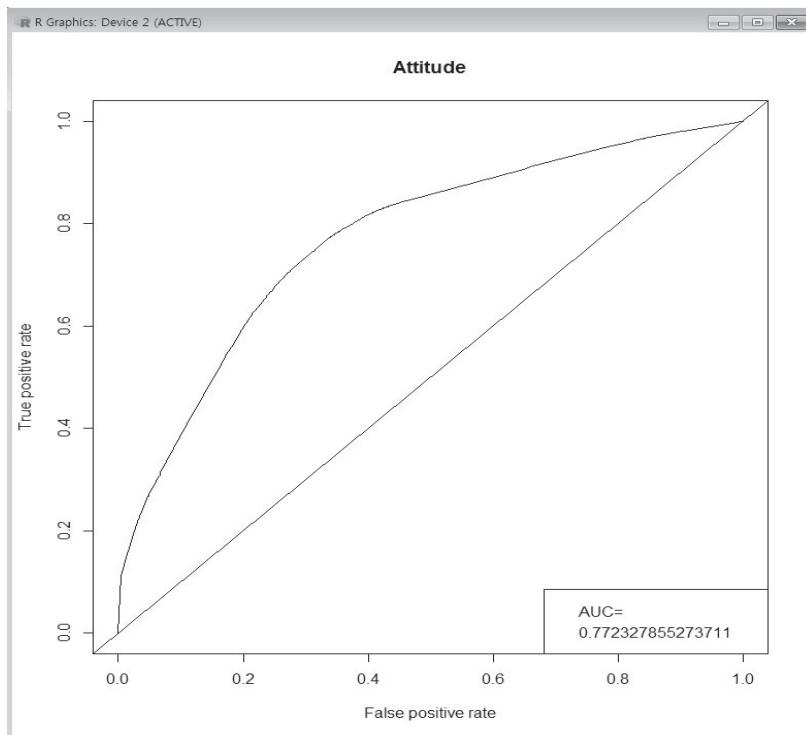
> install.packages('ROCR')
  - Install the package that generates the ROC curve.
> library(ROCR)
> par(mfrow=c(1,1))
  - The par () function is used to query or set the graphics argument.
  - mfrow = c (1,1): used to set one (1*1) plot on one screen.
> pred_obs=read.table('cyberbullying_attitude_neuralnet.txt',header=T)
  - Open the data file containing the predicted probability value
    (p_Attitude) and then assign the pred-obs object.
> PO_c=prediction(pred_obs$p_Attitude, pred_obs$Attitude)
  - Predict the estimate of tdata’s Attitude using the real group and the
    predictive group.
> PO_cf=performance(PO_c, "tpr", "fpr")
  - Create a tpr(true positive rate) and a fpr(false positive rate) of the
    ROC curve.
> auc_PO=performance(PO_c,measure="auc")
  - Evaluate the performance of the AUC curve.
> auc_PO@y.values: Calculate the AUC statistics.
> plot(PO_cf,main='Attitude')
  - Draw a ROC curve with Title as Attitude.
> legend('bottomright',legend=c('AUC=', auc_PO@y.values))

```

- Include the AUC statistic in the legend.
- > abline(a=0, b= 1): Draw the baseline of the ROC curve.

```
R Console
> # ROC curve
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('ROCR')
Warning: package 'ROCR' is in use and will not be installed
> library(ROCR)
> par(mfrow=c(1,1))
> pred_obs = read.table('cyberbullying_attitude_neuralnet.txt',header=T)
> PO_c=prediction(pred_obs$p_Attitude, pred_obs$Attitude)
> PO_cf=performance(PO_c, "tpr", "fpr")
> auc_PO=performance(PO_c,measure="auc")
> auc_PO@y.values
[1]
[1] 0.7723278553

> plot(PO_cf,main='Attitude')
> legend('bottomright',legend=c('AUC=', auc_PO@y.values))
> abline(a=0, b= 1)
> |
```



Analysis: The receiver operation characteristic (ROC) curve's performance was 77.23% (moderately accurate).

Cyber bullying Type Prediction Model

The neural network model for predicting types (Perpetrator, Victim, Bystander, Complex) for cyber bullying is as follows.

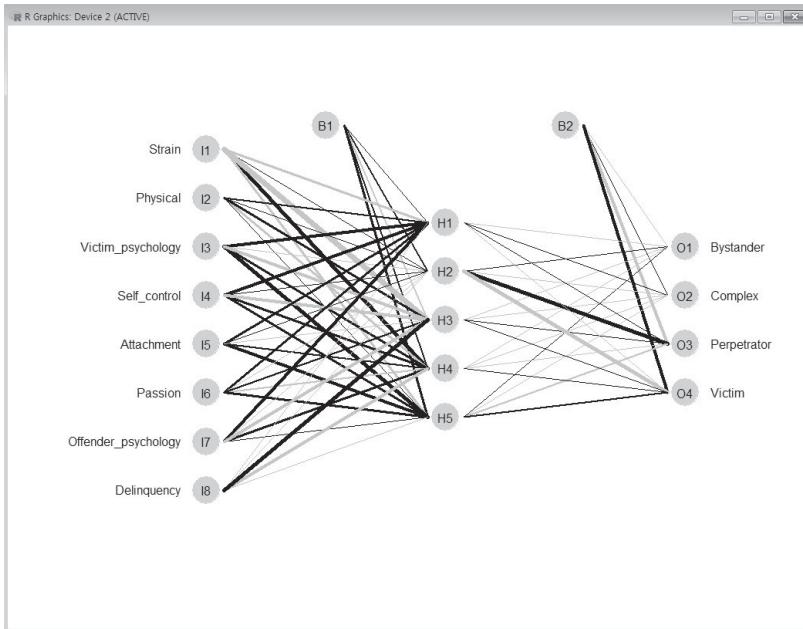
```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("nnet")
> library(nnet)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_type.txt',header=T,sep=",")
> p_output=read.table('p_output_type_n.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
> p_output_vars = c(colnames(p_output))
> tr.nnet = nnet(form, data=tdata, size=5)
> p=predict(tr.nnet, tdata, type='raw')
> dimnames(p)=list(NULL,c(p_output_vars))
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'cyberbullying_type_neural.txt')
> mydata1=read.table('cyberbullying_type_neural.txt',header=T)
> attach(mydata1)
> mean(mydata1$p_Perpetrator)
> mean(mydata1$p_Victim)
> mean(mydata1$p_Bystander)
> mean(mydata1$p_Complex)
```

A screenshot of an R console window titled "R Console". The window contains R code for building a neural network model. The code includes reading data from a file, defining variables, creating a formula, fitting a neural network (nnet) with 5 hidden units, and writing the predicted values to a text file. The console also shows the mean prediction probability for each category: perpetrator (12.17%), victim (70.82%), bystander (9.37%), and complex (7.65%).

```
> p_output=read.table('p_output_type_n.txt',header=T,sep=",")  
Warning message:  
In read.table("p_output_type_n.txt", header = T, sep = ",") :  
  incomplete final line found by readTableHeader on 'p_output_type_n.txt'  
>  
> # neural networks modeling  
>  
> input_vars = c(colnames(input))  
> output_vars = c(colnames(output))  
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',  
+ paste(input_vars, collapse = '+')))  
> form  
Type ~ Strain + Physical + Victim_psychology + Self_control +  
  Attachment + Passion + Offender_psychology + Delinquency  
> p_output_vars = c(colnames(p_output))  
>  
> tr.nnet = nnet(form, data=tdata, size=5)  
# weights:  69  
initial value 75933.119523  
iter  10 value 53597.013578  
iter  20 value 52293.395797  
iter  30 value 51877.912304  
iter  40 value 51629.095446  
iter  50 value 51513.499836  
iter  60 value 51443.829469  
iter  70 value 51407.568550  
iter  80 value 51384.060956  
iter  90 value 51372.761978  
iter 100 value 51365.090019  
final value 51365.090019  
stopped after 100 iterations  
> p=predict(tr.nnet, tdata, type='raw')  
> dimnames(p)=list(NULL,c(p_output_vars))  
> pred_obs = cbind(tdata, p)  
>  
> write.matrix(pred_obs,'cyberbullying_type_neural.txt')  
> mydata1=read.table('cyberbullying_type_neural.txt',header=T)  
> #attach(mydata1)  
> mean(mydata1$p_Perpetrator)  
[1] 0.1217284021  
> mean(mydata1$p_Victim)  
[1] 0.7081576808  
> mean(mydata1$p_Bystander)  
[1] 0.09365991126  
> mean(mydata1$p_Complex)  
[1] 0.07645400584  
> |
```

Analysis: In the neural network model using “nnet,” the dependent variable’s mean prediction probability was 12.17% for perpetrator, 70.82% for victim, 9.37% for bystander, and 7.65% for complex.

```
> install.packages('NeuralNetTools')
> library(NeuralNetTools)
> plotnet(tr.nnet)
```

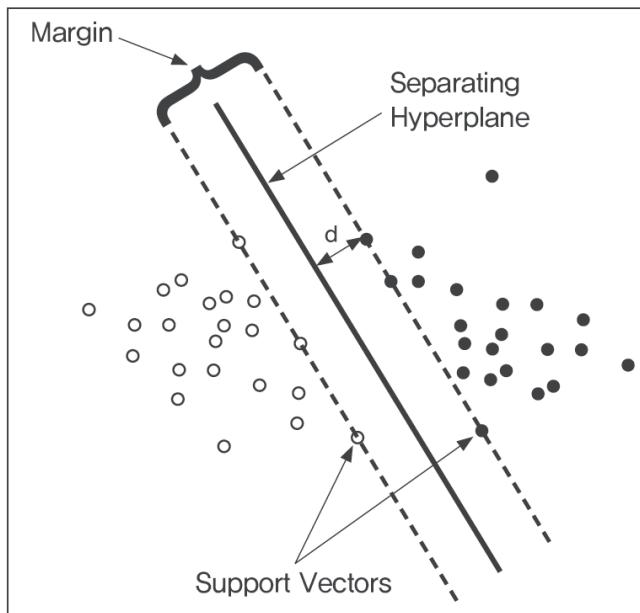


Support Vector Machine Model

The support vector machine (SVM), proposed by Cortes and Vapnic (1995), uses supervised learning. It can be used for both classification and regression. Logistic regression estimates the conditional probability of an output value for a given input value. By comparison, the SVM does not estimate a probability, but directly predicts the classification results. Thus, it has a higher overall predictive power than probability estimation methods, if we look at classification efficiency in big data (populations) by itself. As shown in Fig. 8, the SVM creates a model that maximizes the margin (and minimizes misclassifications) between data on two support vectors that define the boundary of the two groups ($y = 1$, $y = -1$). The margins of the two groups show the distance (d) between the two support vectors, and the two groups' classification equation is as follows.

$$f(x) = w \cdot x + w_0 \quad (7)$$

Here, w is the estimation parameter and x is the input value. ‘·’ is the vector symbol and refers to $(w_1x_1 + w_2x_2 + \dots + w_nx_n)$. w_0 is the bias. $f(x)$ is the classification function.



Reference: <https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf>
Fig. 8. Support Vector Machine Classification (linear separable case)

Cyber bullying Risk (Negative, Positive) Prediction Model

The SVM model that predicts emotions (Negative, Positive) about cyber bullying is as follows. In R, SVM models use the “e1071” package.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> library(e1071)
> library(caret)
> library(kernlab)
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
```

```
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
> form
> svm.model=svm(form,data=tdata,kernel='radial')
> summary(svm.model)
> p=predict(svm.model,tdata)
> mean(p)
```



The screenshot shows the R Console window with the following content:

```
R Console

> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> # SVM modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> # If you run it, you'll have to spend about 'time: 2.75 hours'
> svm.model=svm(form,data=tdata,kernel='radial')
> summary(svm.model)

Call:
svm(formula = form, data = tdata, kernel = "radial")

Parameters:
  SVM-Type:  eps-regression
  SVM-Kernel: radial
    cost: 1
    gamma: 0.125
  epsilon: 0.1

Number of Support Vectors:  84824

>
> # If you run it, you'll have to spend about 'time: 15 minutes'
> p=predict(svm.model,tdata)
> mean(p)
[1] 0.5729908312
> mydata=cbind(tdata, p)
> write.matrix(mydata,'cyberbullying_attitude_SVM.txt')
> |
```

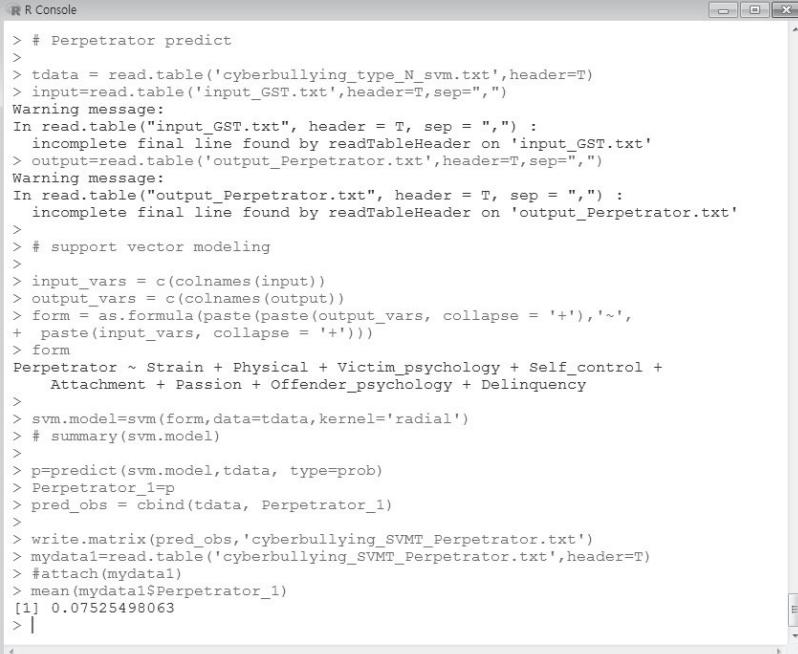
Analysis: In the support vector machine, the dependent variable's mean prediction probability was 57.30% for positive and 42.70% for negative.

Cyber bullying Type Prediction Model

The support vector machine model for predicting types (Perpetrator, Victim, Bystander, Complex) for cyber bullying is as follows. To use a support vector machine model to predict dependent variables (labels) with multiple categories, e.g., the cyber bullying types, the prediction model must be developed according to the types.

■ Perpetrator prediction

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages(e1071)
> library(e1071)
> install.packages('caret')
> library(caret)
> install.packages('kernlab')
> library(kernlab)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_type_N_svm.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_Perpetrator.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> svm.model=svm(form,data=tdata,kernel='radial')
> p=predict(svm.model,tdata, type=prob)
> Perpetrator_1=p
  - Assigns the perpetrator's prediction probability to 'Perpetrator_1'.
> pred_obs = cbind(tdata, Perpetrator_1)
> write.matrix(pred_obs,'cyberbullying_SVMT_Perpetrator.txt')
> mydata1=read.table('cyberbullying_SVMT_Perpetrator.txt',header=T)
> attach(mydata1)
> mean(mydata1$Perpetrator_1)
```



```

R R Console
> # Perpetrator predict
>
> tdata = read.table('cyberbullying_type_N_svm.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_Perpetrator.txt',header=T,sep=",")
Warning message:
In read.table("output_Perpetrator.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Perpetrator.txt'
>
> # support vector modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Perpetrator ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> svm.model=svm(form,data=tdata,kernel='radial')
> # summary(svm.model)
>
> p=predict(svm.model,tdata, type=prob)
> Perpetrator_1=p
> pred_obs = cbind(tdata, Perpetrator_1)
>
> write.matrix(pred_obs,'cyberbullying_SVMT_Perpetrator.txt')
> mydata1=read.table('cyberbullying_SVMT_Perpetrator.txt',header=T)
> #attach(mydata1)
> mean(mydata1$Perpetrator_1)
[1] 0.0725498063
> |

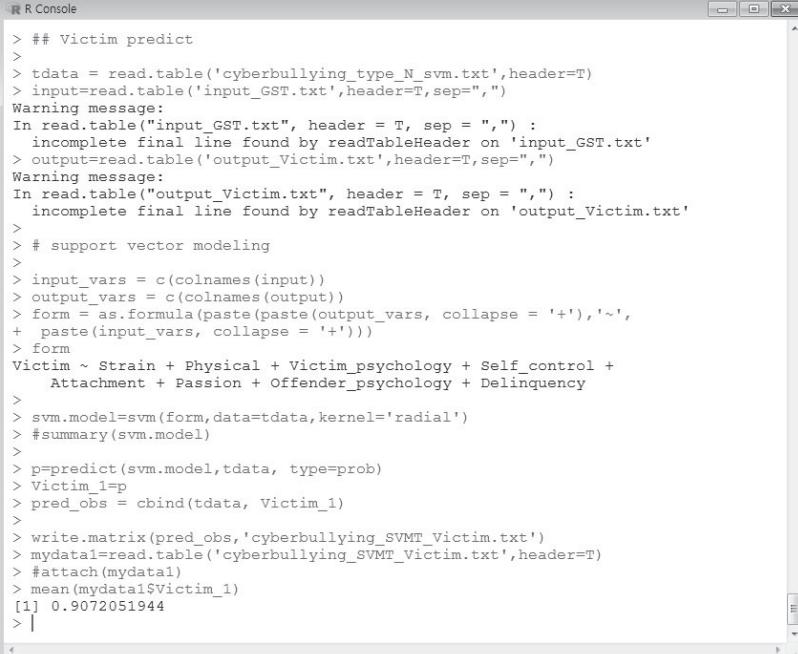
```

■ Victim Prediction

```

> tdata = read.table('cyberbullying_type_N_svm.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_Victim.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
> form
> svm.model=svm(form,data=tdata,kernel='radial')
> p=predict(svm.model,tdata, type=prob)
> Victim_1=p
  - Assign the victim's prediction probability to 'Victim_1'.
> pred_obs = cbind(tdata, Victim_1)
> write.matrix(pred_obs,'cyberbullying_SVMT_Victim.txt')
> mydata1=read.table('cyberbullying_SVMT_Victim.txt',header=T)
> attach(mydata1)
> mean(mydata1$Victim_1)

```

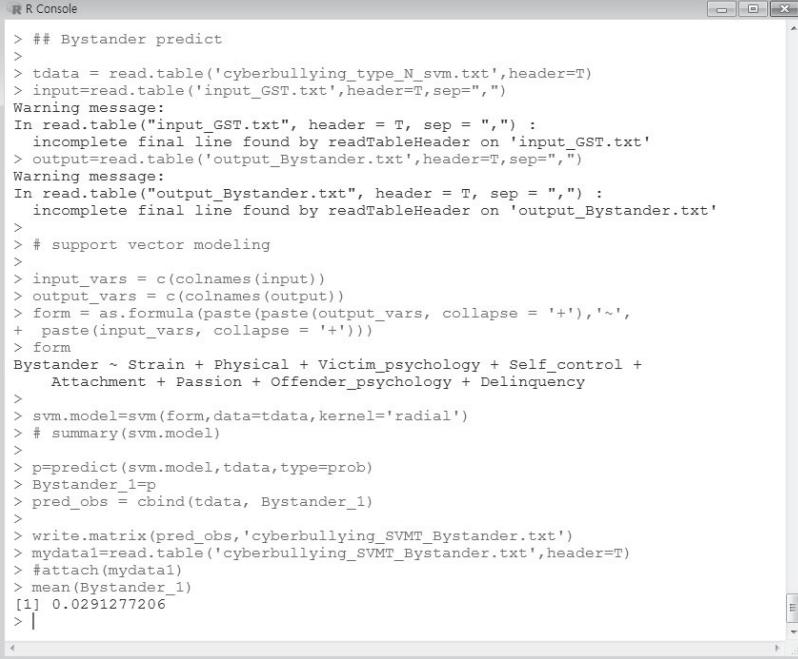


The screenshot shows the R console window with the following R code:

```
> ## Victim predict
>
> tdata = read.table('cyberbullying_type_N_svm.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_Victim.txt',header=T,sep=",")
Warning message:
In read.table("output_Victim.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Victim.txt'
>
> # support vector modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Victim ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> svm.model=svm(form,data=tdata,kernel='radial')
> #summary(svm.model)
>
> p=predict(svm.model,tdata, type=prob)
> Victim_1=p
> pred_obs = cbind(tdata, Victim_1)
>
> write.matrix(pred_obs,'cyberbullying_SVMT_Victim.txt')
> mydata1=read.table('cyberbullying_SVMT_Victim.txt',header=T)
> #attach(mydata1)
> mean(mydata1$Victim_1)
[1] 0.9072051944
> |
```

■ Bystander Prediction

```
> tdata = read.table('cyberbullying_type_N_svm.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_Bystander.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> svm.model=svm(form,data=tdata,kernel='radial')
> p=predict(svm.model,tdata,type=prob)
> Bystander_1=
  - Assign the bystander's prediction probability to 'Victim_1'.
> pred_obs = cbind(tdata, Bystander_1)
> write.matrix(pred_obs,'cyberbullying_SVMT_Bystander.txt')
> mydata1=read.table('cyberbullying_SVMT_Bystander.txt',header=T)
> attach(mydata1)
> mean(mydata1$Bystander_1)
```



The screenshot shows the R Console window with the following R script:

```

R R Console
> ## Bystander predict
>
> tdata = read.table('cyberbullying_type_N_svm.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_Bystander.txt',header=T,sep=",")
Warning message:
In read.table("output_Bystander.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Bystander.txt'
>
> # support vector modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Bystander ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> svm.model=svm(form,data=tdata,kernel='radial')
> # summary(svm.model)
>
> p=predict(svm.model,tdata,type=prob)
> Bystander_1=p
> pred_obs = cbind(tdata, Bystander_1)
>
> write.matrix(pred_obs,'cyberbullying_SVMT_Bystander.txt')
> mydata1=read.table('cyberbullying_SVMT_Bystander.txt',header=T)
> #attach(mydata1)
> mean(Bystander_1)
[1] 0.0291277206
> |

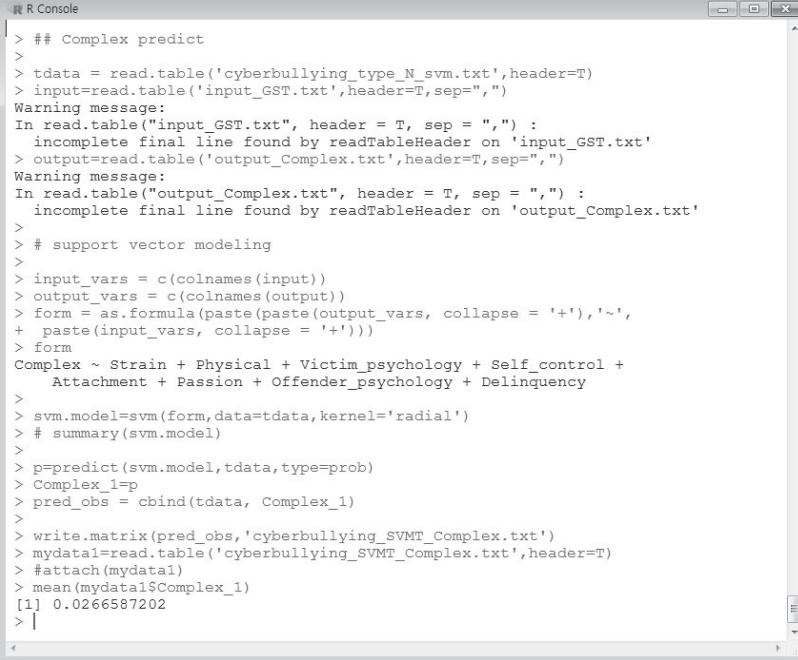
```

■ Complex Prediction

```

> tdata = read.table('cyberbullying_type_N_svm.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_Complex.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> svm.model=svm(form,data=tdata,kernel='radial')
> p=predict(svm.model,tdata,type=prob)
> Complex_1=p
  - The complex prediction probability is assigned to 'Complex_1'.
> pred_obs = cbind(tdata, Complex_1)
> write.matrix(pred_obs,'cyberbullying_SVMT_Complex.txt')
> mydata1=read.table('cyberbullying_SVMT_Complex.txt',header=T)
> attach(mydata1)
> mean(mydata1$Complex_1)

```



```

R Console
> ## Complex predict
>
> tdata = read.table('cyberbullying_type_N_svm.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_Complex.txt',header=T,sep=",")
Warning message:
In read.table("output_Complex.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Complex.txt'
>
> # support vector modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Complex ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> svm.model=svm(form,data=tdata,kernel='radial')
> # summary(svm.model)
>
> p=predict(svm.model,tdata,type=prob)
> Complex_1=p
> pred_obs = cbind(tdata, Complex_1)
>
> write.matrix(pred_obs,'cyberbullying_SVMT_Complex.txt')
> mydata1=read.table('cyberbullying_SVMT_Complex.txt',header=T)
> #attach(mydata1)
> mean(mydata1$Complex_1)
[1] 0.0266587202
> |

```

■ File merge (combines prediction probabilities into one file)

```

> mydata1=read.table('cyberbullying_SVMT_Perpetrator.txt',header=T)
> mydata2=read.table('cyberbullying_SVMT_Victim.txt',header=T)
> mydata3=read.table('cyberbullying_SVMT_Bystander.txt',header=T)
> mydata4=read.table('cyberbullying_SVMT_Complex_psychology.txt',
  header=T)
> mydata5=cbind(mydata1, mydata2$Victim_1, mydata3$Bystander_1,
  mydata4$Complex_1)
> write.matrix(mydata5,'cyberbullying_SVMT_total.txt')
> mydata5=read.table('cyberbullying_SVMT_total.txt',header=T)
> attach(mydata5)
> mean(mydata5$Perpetrator_1)
> mean(mydata2$Victim_1)
> mean(mydata3$Bystander_1)
> mean(mydata4$Complex_1)

```



```

> # combine into one file
>
> mydata1=read.table('cyberbullying_SVMT_Perpetrator.txt',header=T)
> mydata2=read.table('cyberbullying_SVMT_Victim.txt',header=T)
> mydata3=read.table('cyberbullying_SVMT_Bystander.txt',header=T)
> mydata4=read.table('cyberbullying_SVMT_Complex.txt',header=T)
>
> mydata5=cbind(mydata1, mydata2$Victim_1, mydata3$Bystander_1, mydata4$Complex_1)
> write.matrix(mydata5,'cyberbullying_SVMT_total.txt')
> mydata5=read.table('cyberbullying_SVMT_total.txt',header=T)
> #attach(mydata5)
> mean(mydata5$Perpetrator_1)
[1] 0.07525498063
> mean(mydata2$Victim_1)
[1] 0.9072051944
> mean(mydata3$Bystander_1)
[1] 0.0291277206
> mean(mydata4$Complex_1)
[1] 0.0266587202
> |

```

Analysis: In the support vector machine model, the dependent variable's mean prediction probability was 7.53% for perpetrator, 90.72% for victim, 2.91% for bystander, and 2.67% for complex.

Association Analysis

Association analysis searches for meaningful relationships between variables in large-scale databases. It does not require a particular statistical process, and it finds association rules that are hiding in big data.

Association analysis is the analysis technique used in the “men buy beer and diapers together” shopping cart analysis example, and it can be used to expand the shopping cart analysis to social data keywords (words).

Association analysis for social big data discovers the correlation between two or more keywords included in a single online document (transaction). It finds the conditions and association rules for certain sets of keywords that appear at the same time. The association rule’s evaluation measurement in all documents is expressed in terms of support, confidence, and lift.

Support is the ratio of data that corresponds to the association rule ($X \rightarrow Y$) to all documents ($s = \frac{n(X \cup Y)}{N}$). Confidence is the ratio of documents that contain the keyword X and keyword Y to all documents that contain the keyword X ($c = \frac{n(X \cup Y)}{n(X)}$). Lift is the ratio of the increase in the probability of keyword Y when keyword X is given, compared to the probability of Y when X is not given ($l = \frac{c(X \rightarrow Y)}{s(Y)}$).

As the lift increases, whether or not X occurs has a larger effect on whether or not Y occurs. As such, the support can be used to remove rules

that do not occur often, and the confidence can be used to understand the degree of association between words. Lift shows the ratio at which Y appears more when X appears than when X does not appear in the association rule ($X \rightarrow Y$).

The association analysis process consists of creating a frequent item set that satisfies the minimum support specified by the researcher, and then creating a minimum confidence standard for this set and setting a lift of 1 or more as a rule (Park, 2013).

Social big data association analysis uses the Apriori algorithm to find the association rules of keywords that appear in documents (binomial data). Binomial data are data measured by whether or not a keyword appears in a document. The Apriori algorithm was proposed in 1994 by R. Agrawal and R. Srikant (1994) and is used in association rule learning.

Association rules can be found by using the Apriori function in R's "arules" package. In the association analysis of social big data, one method finds rules between keywords (e.g., related to cyber bullying), and another finds rules between cyber bullying keywords and dependent variables (e.g., the types in cyber bullying [perpetrator, victim, bystander, complex]).

Analysis of Associations between Keywords

Below is the association analysis process between delinquency factor keywords (Access_entertainment_facilities ~ Student_violence) regarding cyber bullying.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017"): Set the working directory.
> install.packages("arules"): 'Install the 'arules' package.
> library(arules): Loads the 'arules' package.
> rm(list=ls())
> cyber_bullying=read.table(file='cyberbullying_type_association.txt',
  header=T)
  - Assign cyberbullying data (delinquency factor) files to the
    cyber_bullying variable.
> attach(cyber_bullying)
> cyber_bullying_asso=cbind(Access_entertainment_facilities,Smoking,
  Drinking, Drug, Run_away, Gambling, Crime, Pregnant, Sexual_violence,
  Sex, Absence_with_out_leave, Student_violence)
  - Assign 12 variables of cyberbullying to the cyberbullying_asso vector.
> cyber_bullying_trans=as.matrix(cyber_bullying_asso,"Transaction")
```

- Convert the cyber_bullying_asso variable to a matrix file with values 0 and 1. And then assign them to the cyber_bullying_trans variable.

```
> rules1=apriori(cyber_bullying_trans,parameter=list(supp=0.01,conf=0.01,target="rules"))
- Find the rule with a support of 0.01 and a confidence of 0.01 or higher, and then assign them to the rule1 variable.
```

> summary(rules1)

- Summary of association rules and output on the screen.

```
> rules.sorted=sort(rules1, by="confidence"): Sort by confidence.
```

> inspect(rules.sorted)

- Sort the confidence in order and display on the screen.
- The inspect() function returns the values of lhs, rhs, support, confidence, lift, and count.
- lhs (left-hand-side) means antecedent, and rhs (right-hand-side) means consequent.

```
> rules.sorted=sort(rules1, by="lift"): Sort by lift.
```

> inspect(rules.sorted)

```
> write(rules.sorted, file = "association_result.csv", sep = ",")
```

- Save the result.

The screenshot shows the R console window with the following text output:

```

R Console
> rules.sorted=sort(rules1, by="lift")
> inspect(rules.sorted)
   lhs                  rhs          support    confidence      lift       count
[1] {Student_violence} => {Absence_without_leave} 0.06854339756 0.89705031810 4.3568264859 1551
[2] {Absence_without_leave} => {Student_violence} 0.06854339756 0.33290405666 4.3568264859 1551
[3] {Sex}                  => {Sexual_violence} 0.0108181899 0.093028331052 2.2077378509 227
[4] {Sexual_violence}      => {Sex}             0.0039318899 0.093028331052 2.2077378509 227
[5] {Drinking}              => {Smoking}        0.02081491957 0.23065621539 4.204916100 471
[6] {Run_away}               => {Drinking}        0.02081491957 0.19890202703 2.204916100 471
[7] {Run_away,Crime}        => {Smoking}        0.0106947451 0.22060164084 2.1080126389 242
[8] {Smoking,Crime}         => {Run_away}        0.0106947451 0.31842105263 1.59867574742 242
[9] {Run_away}               => {Absence_without_leave} 0.0605865123 0.30419347681 1.4774178994 1371
[10] {Absence_without_leave} => {Run_away}        0.0605865123 0.29426915647 1.4774178994 1371
[11] {Crime}                 => {Sexual_violence} 0.06275410998 0.05911855949 1.1217195594 1420
[12] {Sexual_violence}       => {Crime}           0.00039318899 0.05911855949 1.1217195594 1420
[13] {Smoking}                => {Run_away}        0.02090330564 0.19746621261 1.0028547934 473
[14] {Run_away}                => {Smoking}        0.02090330564 0.10494785889 1.0028547934 473
[15] {}                      => {Student_violence} 0.07640975782 0.07640975782 1.0000000000 1729
[16] {}                      => {Access_entertainment_facilities} 0.01316952448 0.01316952448 1.0000000000 298
[17] {}                      => {Gambling}        0.01595368570 0.01595368570 1.0000000000 361
[18] {}                      => {Pregnant}        0.04056920629 0.04056920629 1.0000000000 918
[19] {}                      => {Sex}             0.00039318899 0.00039318899 1.0000000000 9645
[20] {}                      => {Drugs}           0.04538624713 0.04538624713 1.0000000000 1027
[21] {}                      => {Drinking}        0.09024217783 0.09024217783 1.0000000000 2047
[22] {}                      => {Smoking}        0.10464910730 0.10464910730 1.0000000000 2368
[23] {}                      => {Sexual_violence} 0.10654940783 0.10654940783 1.0000000000 2411
[24] {}                      => {Run_away}        0.19917800955 0.19917800955 1.0000000000 4507
[25] {}                      => {Absence_without_leave} 0.20589535089 0.20589535089 1.0000000000 4659
[26] {}                      => {Crime}            0.52874574508 0.52874574508 1.0000000000 11881
[27] {Smoking,Run_away}     => {Sex}             0.0106947451 0.51162060699 0.9742271582 472
[28] {Sex}                      => {Sex}             0.0106947451 0.03062311169 0.9058964263 459
[29] {Sex}                      => {Crime}           0.020284603135 0.47564766839 0.9058964263 459
[30] {Drug}                     => {Crime}           0.02121265689 0.46738072055 0.8901515819 480
[31] {Crime}                     => {Drug}            0.02121265689 0.04040063968 0.8901515819 480
[32] {Pregnant}                 => {Crime}           0.01418596429 0.34967320261 0.6659713180 321
[33] {Crime}                     => {Pregnant}        0.01418596429 0.02701792778 0.6659713180 321
[34] {Drinking}                  => {Crime}           0.0106947451 0.03062311169 0.6131646400 760
[35] {Crime}                     => {Drinking}        0.03358676764 0.0639677949 0.6125872711 760
[36] {Drinking}                  => {Run_away}        0.01056213541 0.11704211557 0.58762569134 239
[37] {Run_away}                   => {Drinking}        0.01056213541 0.053028622214 0.58762569134 239
[38] {Smoking}                   => {Absence_without_leave} 0.01246243598 0.11908783784 0.5783901255 282
[39] {Absence_without_leave}    => {Smoking}        0.01246243598 0.06052801030 0.5783901255 282
[40] {Drinking}                  => {Crime}           0.02731129574 0.30264446621 0.5764025740 618
[41] {Crime}                     => {Drinking}        0.02731129574 0.02015823309 0.617640460 618
[42] {Sexual_violence}          => {Sex}             0.0106947451 0.03062311169 0.5016552182 2411
[43] {Run_away}                  => {Sexual_violence} 0.0106947451 0.053028622214 0.5016552182 2411
[44] {Run_away}                  => {Crime}           0.04847975959 0.24339915687 0.4635667134 1097
[45] {Crime}                     => {Run_away}        0.04847975959 0.09233229526 0.4635667134 1097
[46] {Absence_without_leave}   => {Crime}           0.02722290967 0.13221721399 0.2518147562 616
[47] {Crime}                     => {Absence_without_leave} 0.02722290967 0.05184787859 0.2518147562 616
> write(rules.sorted, file = "association_result.csv", sep = ",")
```

Analysis: In the association prediction for delinquency-factor keywords regarding cyber bullying, the association between the two variables of {student violence} => {absence without leave} had a support of 0.069, a confidence of 0.897, and a lift of 4.36. This means that if ‘student violence’ is mentioned in an online document, the probability that ‘absence without leave’ will appear is 89.7%. The probability that ‘absence without leave’ will appear is 4.36 times higher than in documents where ‘student violence’ is not mentioned.

■ Social Network Analysis (SNA) of cyber bullying Delinquency Factor Keyword Association Rules

```
> install.packages("dplyr"): Install the 'dplyr' package for data analysis.  
> library(dplyr): Load the 'dplyr' package.  
> install.packages("igraph")  
  - The 'igraph' package is a graphical representation of linked data.  
> library(igraph)  
> rules = labels(rules1, ruleSep="/", setStart="", setEnd="")  
  - Change the data structure for visualization.  
> rules = sapply(rules, strsplit, "/", USE.NAMES=F)  
  - Change the data structure for visualization.  
> rules = Filter(function(x){!any(x == "")}, rules)  
  - Change the data structure for visualization.  
> rulemat = do.call("rbind", rules)  
  - Change the data structure for visualization.  
> rulequality = quality(rules1): Change the data structure for visualization.  
> ruleeg = graph.edgelist(rulemat, directed=F)  
> ruleeg = graph.edgelist(rulemat[-c(1:16)], directed=F)  
  - Remove '{}' from the association rule results.  
> plot.igraph(ruleeg, vertex.label=V(ruleeg)$name, vertex.label.cex=0.9,  
  vertex.size=20, layout=layout.fruchterman.reingold.grid)  
  - Visualize edgelist.
```

Development of a Cyber Bullying Prediction Model based on Machine Learning

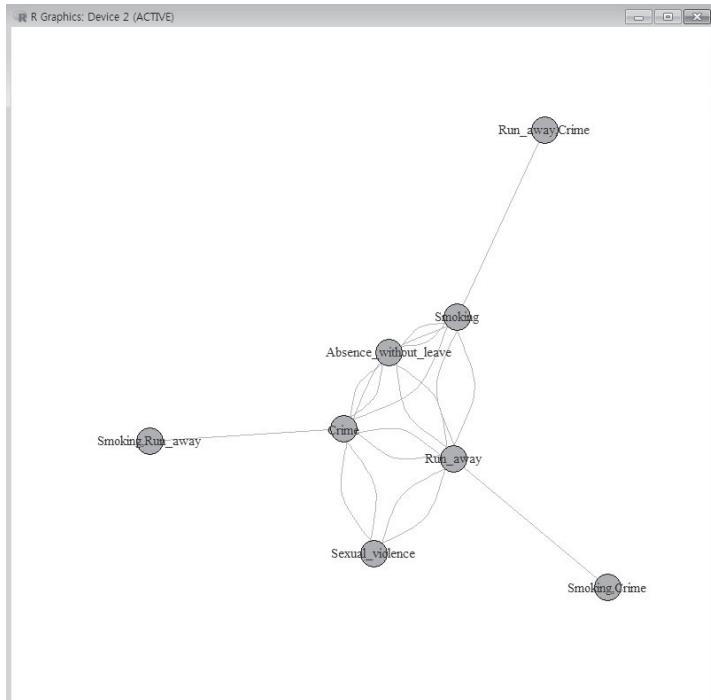
```

R R Console
> ## SNA
>
> install.packages("dplyr")
trying URL 'https://cloud.r-project.org/bin/windows/contrib/3.5/dplyr_0.7.5.zip'
Content type 'application/zip' length 3048303 bytes (2.9 MB)
downloaded 2.9 MB

package 'dplyr' successfully unpacked and MD5 sums checked
Warning: cannot remove prior installation of package 'dplyr'

The downloaded binary packages are in
  C:\Users\SAMSUNG\AppData\Local\Temp\RtmpqMHPwq\downloaded_packages
> library(dplyr)
Error in library(dplyr) : there is no package called 'dplyr'
> install.packages("igraph")
Warning: package 'igraph' is in use and will not be installed
> library(igraph)
> rules = labels(rules1, ruleSep="/", setStart="", setEnd="")
> rules = sapply(rules, strsplit, "/",
+ USE.NAMES=F)
> rules = Filter(function(x) {any(x == "")}, rules)
> rulemat = do.call("rbind", rules)
> rulequality = quality(rules1)
> ruleeg = graph.edgelist(rulemat,directed=F)
>
> # plot for important pairs
>
> ruleeg = graph.edgelist(rulemat[-c(1:16),],directed=F)
> plot.igraph(ruleeg, vertex.label=V(ruleeg)$name, vertex.label.cex=0.9,
+ vertex.size=12, layout=layout.fruchterman.reingold.grid)
Warning message:
In v(graph) : Grid Fruchterman-Reingold layout was removed,
we use Fruchterman-Reingold instead.
> |

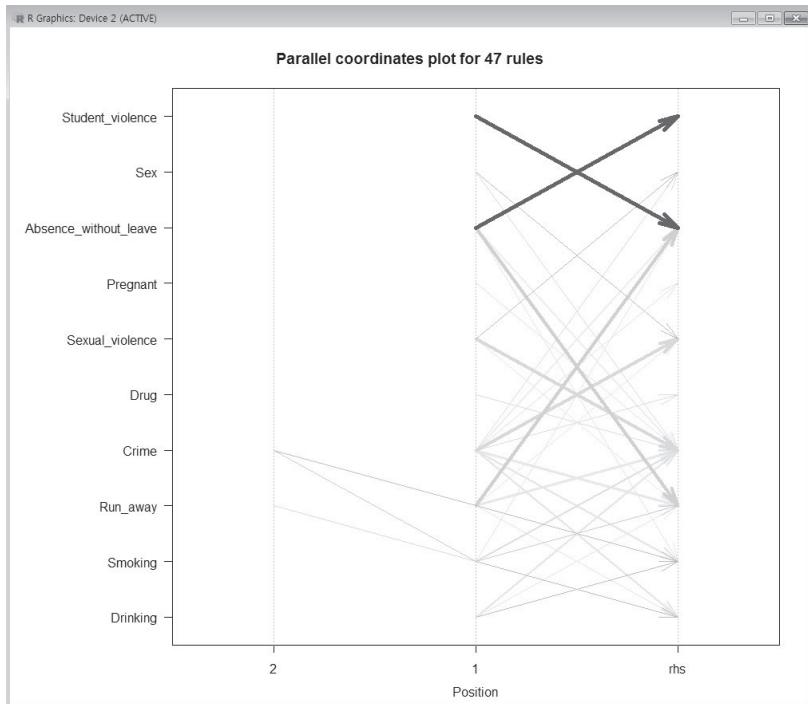
```



Analysis: The results of SNA analysis on association rules are shown in the figure above. The cyber bullying delinquency factor keywords showed that run away, smoking, sexual violence, and crime had a mutual connection with absence without leave.

■ Visualization of Cyber bullying Delinquency Factor Keyword Association Rules

```
> install.packages("arulesViz")
> library(arulesViz)
> plot(rules1, method='paracoord', control=list(reorder=T))
  - Visualization of parallel coordinates plots.
  - In the figure, the line thickness is proportional to the size of the support, and the color tint is proportional to the size of the lift.
  - In the parallel coordinate plot, the terminal point of the x axis is the RHS (right-hand side: consequent) and the combination of the starting point (2) and the midway point (2, 1) is the LHS (left-hand side: antecedent).
  - The intersection of the x and y axes shows the name of the corresponding item (delinquency factor: Access_entertainment_facilities ~ Student_violence). Therefore, when looking at the coordinate, the analysis must reference the association rules.
```



Analysis: The cyber bullying delinquency factor keywords showed that in the first connection stage of LHS (2), crime and Run_away were connected. In the second connection stage of LHS (1), Smoking was connected. In the final connection stage of RHS, Absence_without_leave and Run_away were connected.

Analysis of Association between Keywords and Dependent Variables

Below is the process of analyzing the association between cyber bullying delinquency factor keywords (`access_entertainment_facilities ~ student violence`) and dependent variables [types in cyber bullying (perpetrator, victim, bystander, complex)].

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("arules")
```

```
> library(arules)
> rm(list=ls())
> cyber_bullying=read.table(file='cyberbullying_type_association.txt',
  header=T)
> attach(cyber_bullying)
> cyber_bullying_asso=cbind(Perpetrator,Victim,Bystander,Complex_psychology,
  Access_entertainment_facilities,Smoking,Drinking,Drug,
  Run_away,Gambling,Crime,Pregnant,Sexual_violence,
  Sex,Absence_without_leave,Student_violence)
> trans=as.matrix(cyber_bullying_asso,"Transaction")
> rules1=apriori(trans,parameter=list(supp=0.002,conf=0.01),
  appearance=list (rhs=c("Perpetrator"), default="lhs"), control=list
  (verbose=F))
> rules1=apriori(trans,parameter=list(supp=0.01,conf=0.01),
  appearance=list (rhs=c("Victim"), default="lhs"),control=list
  (verbose=F))
> rules1=apriori(trans,parameter=list(supp=0.002, conf=0.01),
  appearance=list (rhs=c("Bystander"), default="lhs"),control= list
  (verbose=F))
> rules1=apriori(trans,parameter=list(supp=0.002,conf=0.01),
  appearance=list (rhs=c("Complex_psychology"), default="lhs"),
  control=list(verbose=F))
> rules.sorted=sort(rules1, by="confidence")
> inspect(rules.sorted)
> rules.sorted=sort(rules1, by="lift")
> inspect(rules.sorted)
> write(rules.sorted, file = "association_result_de.csv", sep = ", ")
  - Save the result.
```

```

R Console

> rules1=>priori(trans,parameter=list(supp=0.002,conf=0.01), appearance=list
+ (rhs=="Perpetrator"), default="lhs"),control=list(verbose=F)
> rules=.sorts=sort(rules1, by="lift")
> inspect(rules.sorted)

  lhs                                rhs          support   confidence      lift    count
[1] {Absence_without_leave,Student_violence} => {Perpetrator} 0.068057274174 0.99290780142 6.9818265166 1540
[2] {Student_violence}                   => {Perpetrator} 0.068080555772 0.90052053210 6.3321872593 1557
[3] {Absence_without_leave}              => {Perpetrator} 0.073625596606 0.35758746512 2.5144466006 1666
[4] {}                                 => {Perpetrator} 0.142213187202 1.042213187202 1.0000000000 3218
[5] {Drinking}                         => {Perpetrator} 0.008882800078 0.09843290891 0.6921503614 201
[6] {Sex}                               => {Perpetrator} 0.003800601024 0.08911917094 0.6266589811 86
[7] {Smoking}                          => {Perpetrator} 0.009280537388 0.088028243243 0.6266589811 210
[8] {Pregnant}                        => {Perpetrator} 0.01666077426 0.7500000000 1.0985177034 1065
[9] {Run_away,Crime}                  => {Perpetrator} 0.0148486505920 0.07454506989 0.5242123004 336
[10] {Crime}                           => {Perpetrator} 0.038403747569 0.07314199141 0.5143123004 869
[11] {Drug}                            => {Perpetrator} 0.002872547287 0.06329113294 0.4450440953 65
[12] {Smoking,Crime}                  => {Perpetrator} 0.002121265689 0.06315789474 0.4441071604 48
[13] {Sexual_violence}                => {Perpetrator} 0.005877673679 0.05516393243 0.3878953388 133
[14] {Crime,Sexual_violence}          => {Perpetrator} 0.00318198533 0.05070422535 0.3565367344 72
[15] {Run_away,Crime}                 => {Perpetrator} 0.002253844794 0.04649042844 0.3269065925 51
> inspect(rules.sorted)

  lhs                                rhs          support   confidence      lift    count
[1] {Run_away,Absence_without_leave}  => {Victim} 0.0566127806 0.9343544858 1.3685399252 1281
[2] {Access_entertainment_facilities} => {Victim} 0.01060362844 0.8053691275 1.1796163250 240
[3] {Drug,Crime}                     => {Victim} 0.01666077426 0.7854166667 1.1503921505 377
[4] {Run_away}                        => {Victim} 0.15158211066 0.7610383847 1.1146855181 3430
[5] {Crime,Sexual_violence}          => {Victim} 0.04706558246 0.7500000000 1.0985177034 1065
[6] {Drinking,Crime}                 => {Victim} 0.02032879618 0.7443365698 1.0902225320 460
[7] {Drug}                            => {Victim} 0.033675059281 0.74196668939 1.0867516910 762
[8] {Run_away,Crime}                 => {Victim} 0.03592893760 0.7411121240 1.0854997179 813
[9] {Sexual_violence}                => {Victim} 0.07895244683 0.7409521102 1.0854997167 1786
[10] {Crime}                          => {Victim} 0.03695244683 0.7098590102 1.0847697998 8637
[11] {Drinking}                      => {Victim} 0.02047675446 0.7025705207 1.0042120565 607
[12] {Gambling}                      => {Victim} 0.01144559613 0.7174515235 1.0508442666 259
[13] {Crime,Pregnant}               => {Victim} 0.0107601202 0.7102803738 1.0403407534 228
[14] {Crime,Sex}                     => {Victim} 0.01436273643 0.7080610022 1.0370900613 325
[15] {Sex}                           => {Victim} 0.02996287785 0.7025906738 1.0290777242 678
[16] {Smoking,Run_away}              => {Victim} 0.01453950857 0.6955602537 1.0187803366 329
[17] {Smoking,Crime}                => {Victim} 0.02320134347 0.6907894737 1.0117926216 525
[18] {Smoking}                       => {Victim} 0.07203464734 0.6883445944 1.0082116309 1630
[19] {Smoking,Drinking}              => {Victim} 0.01431854340 0.6878980894 1.0075576388 324
[20] {Drinking}                      => {Victim} 0.06191444228 0.6860920666 1.0049123751 1401
[21] {Crime,Absence_without_leave}  => {Victim} 0.01860526781 0.6834415584 1.0010302016 421
[22] {}                             => {Victim} 0.68273820046 0.6827382005 1.0000000000 15449
[23] {Absence_without_leave}        => {Victim} 0.11724412233 0.5694355012 0.8340466387 2653

```

Analysis: When predicting the association between cyber bullying delinquency factors and dependent variables, the association rule with the highest confidence was {Absence_without_leave, Student_violence} => {Perpetrator}. The three variables' association had a support of 0.068, a confidence of 0.9929, and a lift of 6.981. This shows that if Absence_without_leave and Student_violence are mentioned in an online document, the probability of a cyber bullying perpetrator being mentioned is 99.3%. It also shows that the probability of a cyber bullying perpetrator being mentioned is 6.98 higher than in documents that do not mention Absence without leave or Student violence.

```

R Console

> rules1=apriori(trans,parameter=list(supp=0.002,conf=0.01), appearance=list
+ (rhs=c("Bystander"), default="lhs"),control=list(verbose=F))
> rules.sorted=sort(rules1, by="lift")
> inspect(rules.sorted)
      lhs                      rhs          support   confidence      lift      count
[1] {Smoking,Absence_without_leave} => {Bystander} 0.002121265689 0.17021276596 1.7065017581 48
[2] {Smoking,Drinking}              => {Bystander} 0.003491247979 0.16772823779 1.6815926295 79
[3] {Crime,Absence_without_leave} => {Bystander} 0.004065759236 0.14935064935 1.4973444810 92
[4] {Smoking,Crime}               => {Bystander} 0.004905426905 0.14605263158 1.4642795513 111
[5] {Smoking}                    => {Bystander} 0.013169524483 0.12584459459 1.2616798788 298
[6] {Smoking,Run_away}            => {Bystander} 0.002519003005 0.12050739958 1.2081707743 57
[7] {Drinking}                   => {Bystander} 0.010783105083 0.11949069540 1.1979776054 244
[8] {Crime}                      => {Bystander} 0.00765423369 0.11573099907 1.1602840262 1375
[9] {Crime,Sex}                  => {Bystander} 0.002298037829 0.11328976035 1.1358089044 52
[10] {Sexual_violence}           => {Bystander} 0.011799540392 0.11074243053 1.1102701453 267
[11] {Sex}                        => {Bystander} 0.004596075658 0.10777202073 1.0804897142 104
[12] {Run_away,Crime}             => {Bystander} 0.005082199045 0.10483135825 1.0510075208 115
[13] {Pregnant}                  => {Bystander} 0.004242531377 0.10457516340 1.0484389887 96
[14] {Drug}                       => {Bystander} 0.004684461729 0.10321324245 1.0347847808 106
[15] {Crime,Sexual_violence}     => {Bystander} 0.006319604030 0.10070422535 1.0096301335 143
[16] {}                           => {Bystander} 0.099743680396 0.09974368040 1.0000000000 2257
[17] {Drinking,Crime}             => {Bystander} 0.002695771546 0.09870505162 0.9885915333 61
[18] {Run_away}                   => {Bystander} 0.017367862825 0.08719769248 0.8742177162 393
[19] {Absence_without_leave}     => {Bystander} 0.007954746332 0.03863490019 0.3873418350 180
> rules2=apriori(trans,parameter=list(supp=0.002,conf=0.01), appearance=list
+ (rhs=c("Complex"), default="lhs"),control=list(verbose=F))
> rules.sorted=sort(rules2, by="lift")
> inspect(rules.sorted)
      lhs                      rhs          support   confidence      lift      count
[1] {Crime,Absence_without_leave} => {Complex} 0.003181098533 0.11688311688 1.5521309676 72
[2] {Smoking,Run_away}            => {Complex} 0.002238044794 0.10782241015 1.43818107376 51
[3] {Run_away,Crime}             => {Complex} 0.005212659999 0.107566090933 1.4284069656 118
[4] {Crime,Sex}                  => {Complex} 0.002121265999 0.10483116340 1.3885094351 48
[5] {Drinking,Crime}             => {Complex} 0.002441612121 0.1054417474 1.3537103038 63
[6] {Sex}                        => {Complex} 0.004289044412 0.10051810472 1.3461476039 125
[7] {Smoking,Crime}              => {Complex} 0.003938676074 0.10000000000 1.3279342223 76
[8] {Crime,Sexual_violence}     => {Complex} 0.00167024925 0.0995154930 1.3092309727 140
[9] {Smoking}                   => {Complex} 0.010164398081 0.0971293783 1.2998010246 233
[10] {Drinking}                  => {Complex} 0.008661834995 0.0959843290 1.2746088020 194
[11] {Sexual_violence}           => {Complex} 0.009943432915 0.09332227292 1.2392584457 225
[12] {Drug}                      => {Complex} 0.004154145307 0.09152872444 1.2154413008 94
[13] {Pregnant}                 => {Complex} 0.003579635849 0.08823529412 1.2171067109 81
[14] {Crime}                     => {Complex} 0.044193035178 0.08416799933 1.1176957094 1000
[15] {Run_away}                  => {Complex} 0.015379176242 0.07721322387 1.0253408626 348
[16] {}                          => {Complex} 0.075304931943 0.07530493194 1.0000000000 1704
[17] {Absence_without_leave}    => {Complex} 0.007070885628 0.0343213351 0.4560409607 160
> |

```

Analysis: In the cyber bullying delinquency factor and dependent-variable association predictions, the association rule with the highest confidence was {Smoking, Absence_without_leave} => {Bystander}. The association between the three variables had a support of 0.002, a confidence of 0.1702, and a lift of 1.707. This shows that if an online document mentions Smoking and Absence_without_leave, the probability of a cyber bullying bystander being mentioned is 17.02%. It also shows that the probability of a cyber bullying bystander being mentioned is 1.71 times higher than in documents that do not mention Smoking or Absence_without_leave.

Cluster Analysis and Segmentation

Cluster analysis is a technique that classifies a group into several homogenous clusters, based on the similarity of subjects in the group. Cluster analysis omits the machine learning processes of abstraction and generalization and stores the training data as it is; hence, it is called lazy learning or instance-based learning. Instance-based learning machines do

not create a model, but rather perform unsupervised learning, which only groups similar data. Therefore, cluster analysis learns only from independent variables in circumstances with no dependent variables in the data. It produces dependent variables via the independent variables of new data that does not include dependent variables.

Clustering analysis includes nonhierarchical clustering analysis, in which the researcher specifies the number of clusters (K-means clustering analysis), and hierarchical clustering analysis, which groups clusters sequentially among close targets. K-means clustering analysis assigns each object to the cluster with the nearest center (mean); the cluster-connection process is as follows.

First, the number of clusters K is decided upon so that n objects are assigned to K clusters. Second, the mean of each of the K clusters is found. Initially, the data are assigned to the K clusters, and then each cluster's mean is found: $(\bar{x}_i; i = 1, \dots, K)$. Third, the Euclidean distance from each object to the K cluster means is calculated [$d(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_p - y_p)^2}$] to reassign the objects to the closest cluster. Fourth, the reassignment is repeated until convergence occurs (i.e., until the cluster centers have minimal changes).

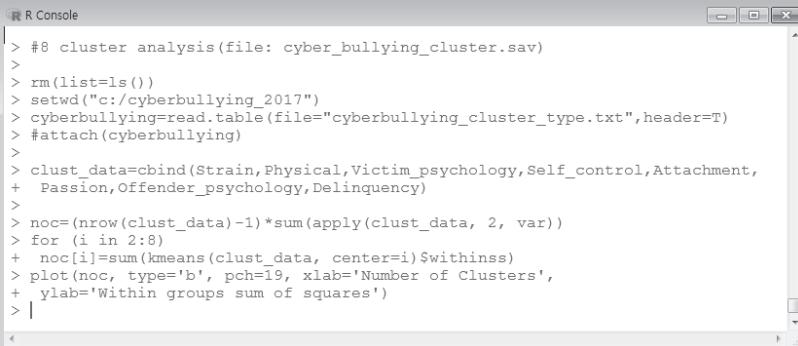
Therefore, in K-means cluster analysis, the number of clusters K must be decided upon beforehand. One method of deciding the number of clusters (selection) is to first select several numbers of clusters and check the results to determine the most suitable number of clusters. When the number of clusters is decided upon, there must be two or more factors that can belong in each of the final clusters. A second method is to use a screen plot to set the number of clusters. A screen plot uses the deviation within clusters (sum of squares within groups) to draw a plot of the clusters. The final number of clusters is determined at the point where a steep slope in the cluster plot becomes gradual or increases.

Cluster Analysis

Research Problem: Perform cluster analysis to segment the GST factors of cyber bullying by cyber bullying types.

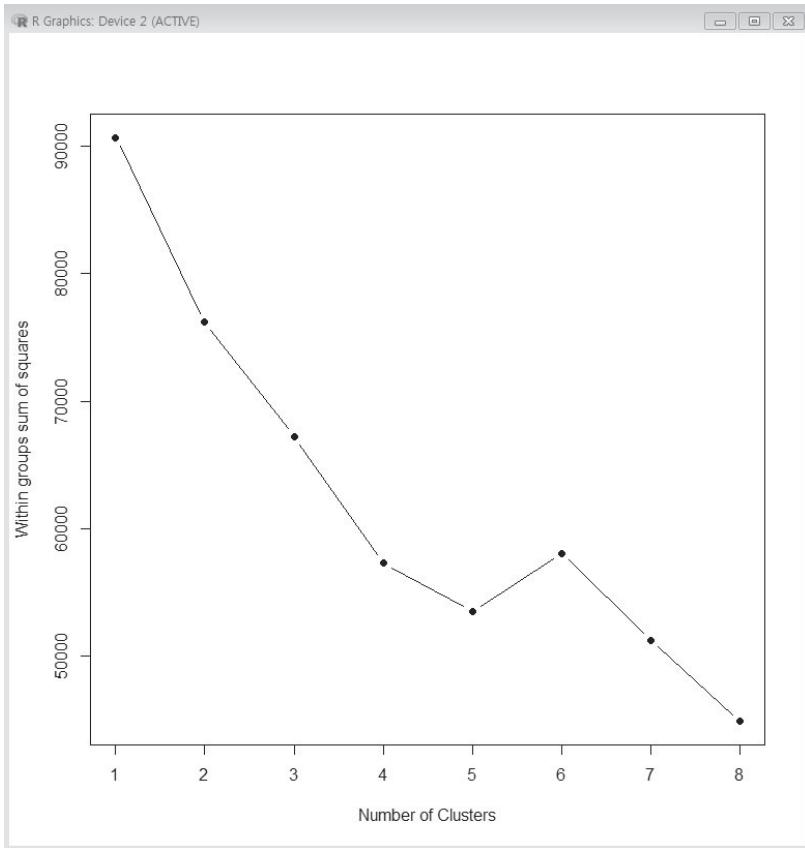
Step 1: Decide on the number of clusters.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017"): Set the working directory.
> cyberbullying=read.table(file="cyberbullying_cluster_type.txt", header=T)
  - Assign the data file to cyber_bullying.
> attach(cyberbullying): Attach the execution data to 'cyberbullying'.
> clust_data=cbind(Strain,Physical,Victim_psychology,Self_control,
  Attachment,Passion,Offender_psychology,Delinquency)
  - GST factors (Strain ~ Delinquency) are combined and assigned to
  clust_data.
> noc=(nrow(clust_data)-1)*sum(apply(clust_data, 2, var))
  - Calculate within groups of squares.
> for (i in 2:8)
  noc[i]=sum(kmeans(clust_data, center=i)$withinss)
> plot(noc, type='b', pch=19, xlab='Number of Clusters',ylab=
  'Within groups sum of squares'): Draw a square chart.
```



The screenshot shows the R Console window with the following R code:

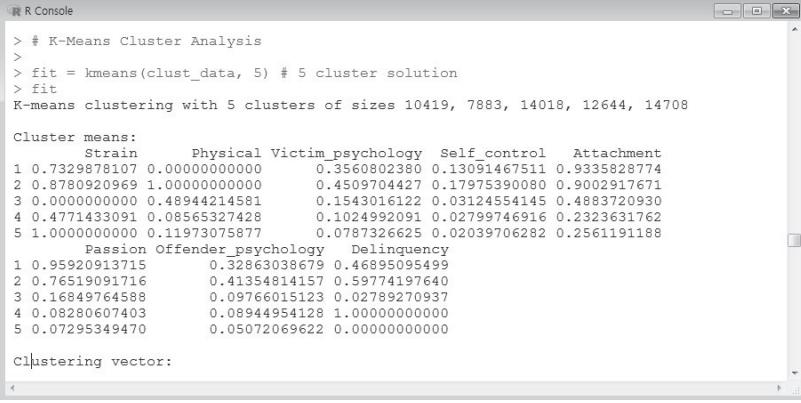
```
R Console
|> # 8 cluster analysis(file: cyber_bullying_cluster.sav)
|> 
|> rm(list=ls())
|> setwd("c:/cyberbullying_2017")
|> cyberbullying=read.table(file="cyberbullying_cluster_type.txt",header=T)
|> #attach(cyberbullying)
|>
|> clust_data=cbind(Strain,Physical,Victim_psychology,Self_control,Attachment,
+   Passion,Offender_psychology,Delinquency)
|>
|> noc=(nrow(clust_data)-1)*sum(apply(clust_data, 2, var))
|> for (i in 2:8)
+   noc[i]=sum(kmeans(clust_data, center=i)$withinss)
|> plot(noc, type='b', pch=19, xlab='Number of Clusters',
+   ylab='Within groups sum of squares')
|> |
```



Analysis: In the screen plot above, the steep slope increases at cluster 5, so the number of clusters is set to five.

Step 2: Perform cluster analysis.

```
> fit = kmeans(clust_data, 5) # 5 cluster solution
  - 5 cluster solution: Create a clust_data object with five clusters.
> fit: Outputs five clusters (fit) to the screen.
```



```

R Console

> # K-Means Cluster Analysis
>
> fit = kmeans(clust_data, 5) # 5 cluster solution
> fit
K-means clustering with 5 clusters of sizes 10419, 7883, 14018, 12644, 14708

Cluster means:
      Strain     Physical Victim_psychology Self_control Attachment
1 0.7329878107 0.000000000000 0.3560802380 0.13091467511 0.9335828774
2 0.8780920969 1.000000000000 0.4509704427 0.17975390080 0.9002917671
3 0.0000000000 0.48944214581 0.1543016122 0.03124554145 0.4883720930
4 0.4771433091 0.08565327428 0.1024992091 0.02799746916 0.2323631762
5 1.0000000000 0.11973075877 0.0787326625 0.02039706282 0.2561191188

      Passion Offender_psychology Delinquency
1 0.95920913715 0.32863038679 0.46895095499
2 0.76519091716 0.41354814157 0.59774197640
3 0.16849764588 0.09766015123 0.02789270937
4 0.08280607403 0.08944954128 1.00000000000
5 0.07295349470 0.05072069622 0.00000000000

```

clustering vector:

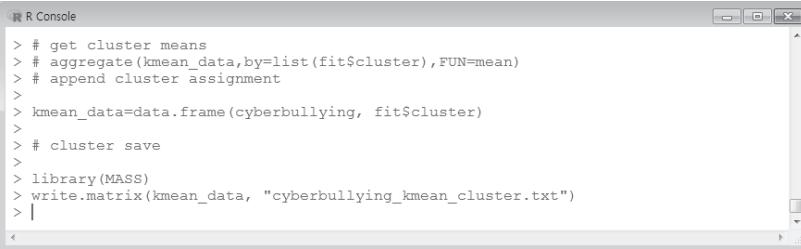
Analysis: Factors with cluster means of 0.3 or greater are included in the cluster. Cluster 1 can classify 10,419 items (Strain, Victim_psychology, Attachment, Passion, Offender_psychology, delinquency). Cluster 2 can classify 7,883 items (Strain, Physical, Victim_psychology, Attachment, Passion, Offender_psychology, delinquency). Cluster 3 can classify 14,018 items (Physical, Attachment). Cluster 4 can classify 12,644 items (Strain, Delinquency). Cluster 5 can classify 14,708 items (Strain).

■ Save the belonging clusters

```

> kmean_data=data.frame(cyberbullying, fit$cluster)
  - Append cluster assignment to 'kmean_data' data.
> library(MASS)
  - Load the MASS package containing the write.matrix () function.
> write.matrix(kmean_data, "cyberbullying_kmean_cluster.txt")
  - Display the kmean_data object to the
    cyberbullying_kmean_cluster.txt file.

```



```

R Console

> # get cluster means
> # aggregate(kmean_data,by=list(fit$cluster),FUN=mean)
> # append cluster assignment
>
> kmean_data=data.frame(cyberbullying, fit$cluster)
>
> # cluster save
>
> library(MASS)
> write.matrix(kmean_data, "cyberbullying_kmean_cluster.txt")
> |

```

Segmentation

The property clusters saved in the cluster analysis can be used to perform segmentation analysis, according to the cyber bullying types. In the cluster analysis' segmentation, a chi-square test is performed to confirm the cyber bullying types [Type (1 = perpetrator, 2 = victim, 3 = bystander, 4 = complex) for the five clusters (fit.cluster) classified in the aforementioned cluster analysis to find each of the clusters' properties.

```
> install.packages('Rcmdr'); library(Rcmdr)
  - Install R Commander package which supports the R graphic environment.
> install.packages('catspec'); library(catspec)
  - Install a package that supports two-way tables.(cross analysis).
> rm(list=ls())
> setwd("c:/cyberbullying_2017"): Set the working directory.
> cyberbullying=read.table(file='cyberbullying_kmean_cluster.txt',
  header=T)
  - Assign the data file to cyber_bullying.
> attach(cyberbullying): Attach execution data as 'cyber_bullying'.
> t1=ftable(cyberbullying[c('fit.cluster','psychological_type')])
  - Assign a two-way tables vector value to the t1 variable for the community type and cyber bullying type.
> ctab(t1,type=c('n','r','c','t'))
  - Display the frequency, row, column and total percentage of the two-way tables on the screen.
> chisq.test(t1)
  - Display the chi-square test statistic of the two-way tables on the screen.
```

```
R Console

> install.packages('Rcmdr'); library(Rcmdr)
Warning: package 'Rcmdr' is in use and will not be installed
> install.packages('catspec'); library(catspec)
Warning: package 'catspec' is in use and will not be installed
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> cyberbullying=read.table(file='cyberbullying_kmean_cluster.txt', header=T)
> #attach(cyberbullying)
>
> t1=ftable(cyberbullying[c('fit.cluster','Type')])
> ctab(ti,type=c('n','r','c','t'))
   Type      1      2      3      4
fit.cluster
1
  Count    794.00  7406.00 1287.00  932.00
  Row %     7.62    71.08  12.35    8.95
  Column %   10.90   17.53  23.06   20.40
  Total %    1.33    12.41   2.16    1.56
2
  Count    625.00  5602.00  817.00  839.00
  Row %     7.93    71.06  10.36   10.64
  Column %   8.58    13.26  14.64   18.36
  Total %    1.05    9.39   1.37    1.41
3
  Count   1246.00 10628.00 1055.00 1089.00
  Row %     8.89    75.82   7.53    7.77
  Column %   17.10   25.16  18.90   23.83
  Total %    2.09    17.81   1.77    1.82
4
  Count   2384.00  8369.00 1146.00  745.00
  Row %     18.85   66.19   9.06    5.89
  Column %   32.72   19.81   20.53   16.31
  Total %    4.00    14.03   1.92    1.25
5
  Count   2236.00 10232.00 1276.00  964.00
  Row %     15.20   69.57   8.68    6.55
  Column %   30.69   24.23   22.86   21.10
  Total %    3.75    17.15   2.14    1.62
> chisq.test(t1)

Pearson's Chi-squared test

data: t1
X-squared = 1437.7523, df = 12, p-value < 2.2204e-16
> |
```

Analysis: Cyber bullying perpetrator was highest in Cluster 4 (Strain, Delinquency) at 32.72%. Victim was highest in Cluster 3 (Physical, Attachment) at 25.16%. Bystander was highest in Cluster 1 (Strain, Victim_psychology, Attachment, Passion, Offender_psychology, Delinquency) at 23.06%. Complex was highest in Cluster 3 (Physical, Attachment) at 23.83%.

MACHINE LEARNING MODEL EVALUATION

Machine learning models are evaluated using the classification accuracy that is demonstrated when the model functions created from the training data are applied to the testing data. Therefore, when evaluating prediction models, they can be tested using a misclassification table for the practical group (actual group) and the prediction group (classification group), as shown in Table 3.

Table 3 Misclassification table [cyber bullying risk (Negative / Positive) prediction]

Prediction group Practical group \	0(Negative)	1(Positive)
0(Negative)	N_{00}	N_{01}
1(Positive)	N_{10}	N_{11}

N: total data number

The classification-model evaluation indexes in Table 3 include “(accuracy) = $(N_{00} + N_{11}) / N$,” which shows the ratio of correctly classified data to all data, and “(error rate) = $(N_{01} + N_{10}) / N$,” which is the misclassification ratio.

“(specificity) = $N_{00} / (N_{00} + N_{01})$ ” is the ratio of correctly classified data to negative documents. “(sensitivity) = $N_{11} / (N_{10} + N_{11})$ ” is the ratio of correctly classified data to positive documents. “(precision) = $N_{11} / (N_{01} + N_{11})$ ” is the ratio of actual positive documents to documents classified as positive. Especially, when evaluating the model of machine learning, sensitivity and specificity are used as an important evaluation index. Sensitivity (true positive rate) refers to be the actual positive, which is the probability of predicting positive. Specificity (true negative rate) is the actual negative, which is the probability of predicting negative. Sensitivity is actually positive, but it aims to minimize "false negative (type-II error (β) refers to the acceptance of H_0 when it is false)" which is predicted by negative. The specificity is actually negative, but it aims to minimize "false positives (type-I error (α) refers to the rejection of H_0 when it is true)" which is predicted by positive. To minimize 'false negative' of Sensitivity and 'false positive' of Specificity, we can select machine

learning algorithm which is highly evaluated for sensitivity, and continuously improve quality of learning data by continuously producing high quality learning data(Proposed in this document).

Table 4 Misclassification table [school cyberbullying types (Perpetrator / Victim / Bystander / Complex) prediction]

Prediction group Practical group	Perpetrator	Victim	Bystander	Complex
Perpetrator	P1	P2	P3	P4
Victim	P5	P6	P7	P8
Bystander	P9	P10	P11	P12
Complex	P13	P14	P15	P16

N: total data number

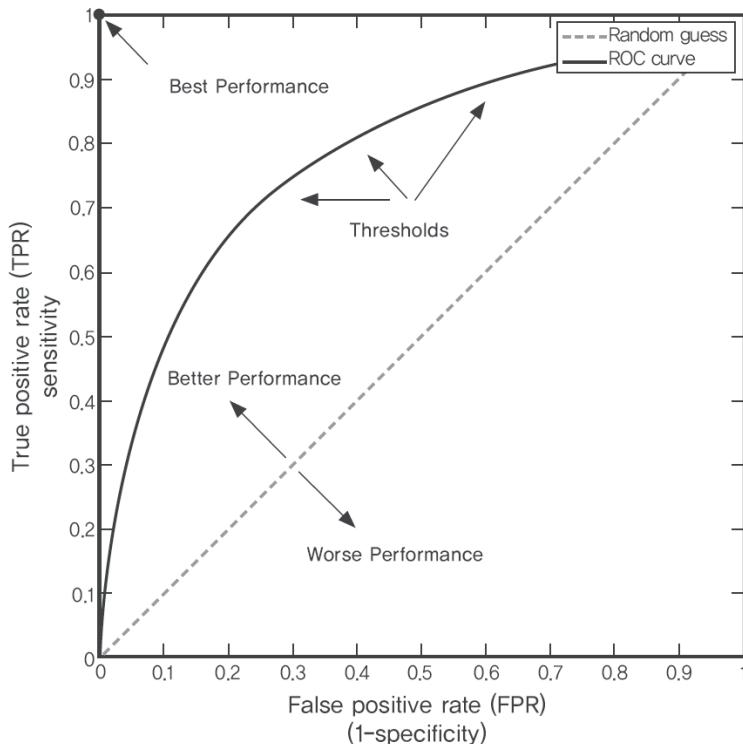
Among the classification model evaluation indexes in Table 4, “accuracy) = (p1+p6+p11+p16) / N” is the ratio of properly classified data to all data.

“(Error rate) = (p2+p3+p4+p7+p8+p12+p5+p9+p10+p13+p14+p15) / N” is the ratio of misclassified data.

“(Upward accuracy) = (p2+p3+p4+p7+p8+p12) / N” is the ratio of properly classified data to upward data.

“(Downward accuracy) = (p5+p9+p10+p13+p14+p15) / N” is the ratio of properly classified data to downward data.

In addition, the machine learning model’s performance tests can be evaluated through ROC (Receiver Operation Characteristic) curves. The ROC shows the relationship between sensitivity and specificity in truncated values. The graph illustrates whether or not the classifier’s performance went over the baseline. The sensitivity and specificity are inversely proportional to each other, and the ROC curve shows an increasing shape (Fig. 9). The ROC curve’s x axis is the FPR (False Positive Rate), and it shows the value of one specificity. The y axis is the TPR (True Positive Rate) and it shows the sensitivity value. ROC uses the AUC (Area Under the Curve), which is the area below the ROC curve, to compare predictive power. As the AUC increases, the classifier can be considered to have better predictive power [less accurate($0.5 < \text{AUC} \leq 0.7$), moderately accurate($0.7 < \text{AUC} \leq 0.9$), highly accurate($0.9 < \text{AUC} < 1$), perfect tests $\text{AUC}=1$] (Greiner et al., 2000: p. 29).



Reference: Hassouna, M., Tarhini, A., Elyas, T. (2015). Customer Churn in Mobile Markets: A Comparison of Techniques. International Business Research, Vol 8(6), pp. 224-237.

Fig. 9. ROC curve

Machine learning goes through a training process so that it can abstract the data (summarize the original data and convert it into an abstract form when training; this creates adaptations through a model that clarifies the patterns between structures) and then generalizes the data (an adjustment process so the abstracted knowledge can be used in practice) (Fig. 10). Therefore, to evaluate a learning machine, the target big data are divided into training data and test data.

The machine learning model's functions are developed from the training data and then applied to the test data to evaluate the practical group and the prediction group. A model with high accuracy is selected and then new data without dependent variables are received as input to predict the dependent variables of the new data.

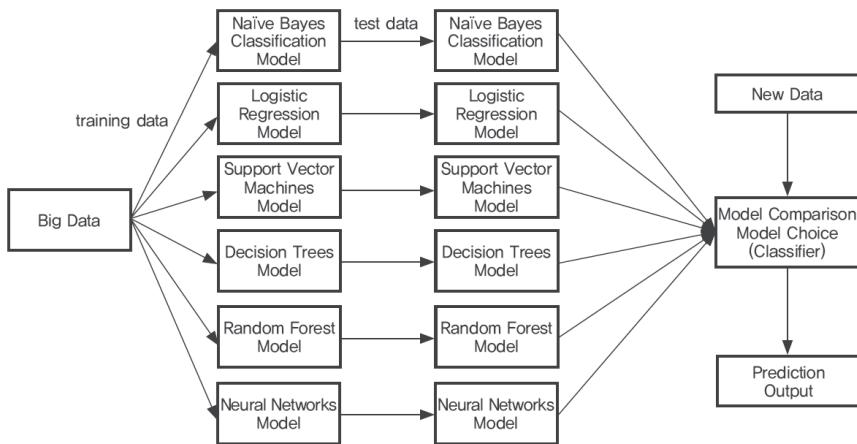


Fig.10. Machine learning process using big data

Machine Learning Model Evaluation Using Misclassification Tables

Below, misclassification tables are used to evaluate the machine learning models that predict the cyber bullying risk and types.

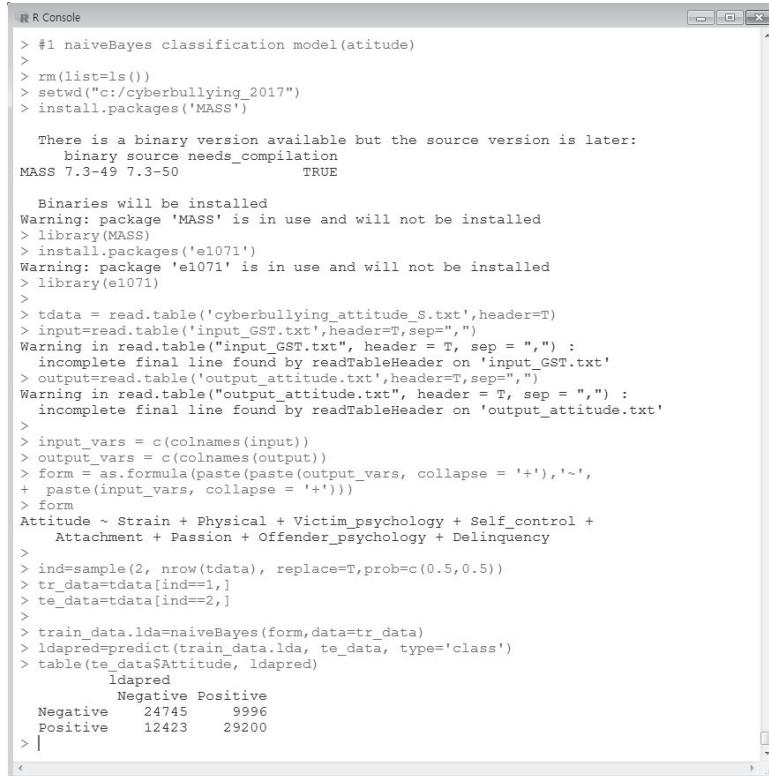
Naïve Bayes Classification Model Evaluation

■ Cyber bullying Risk (Negative, Positive) Prediction Model Evaluation

```

> rm(list=ls()): Initialize all variables..
> setwd("c:/cyberbullying_2017"): Set the working directory.
> install.packages('MASS'): Install the MASS package.
.> library(MASS)
  - Load the MASS package containing the write.matrix () function.
> install.packages('e1071'): Install the e1071 package.
> library(e1071): Load the e1071 package.
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
  - Assign a cyber bullying data file to the tdata object.
  - To evaluate prediction models (model functions) through supervised
    learning, the categories of the dependent variables (Attitude)
    included in the training data must be coded as "<String (character)
    format> (Negative, Positive)."
  
```

```
> input=read.table('input_GST.txt',header=T,sep=",")  
    - Assign an independent variable to the input object as a delimiter (,).  
> output=read.table('output_attitude.txt',header=T,sep=",")  
    - Assign the dependent variable to the output object as a delimiter (,).  
> input_vars = c(colnames(input))  
    - Assign the input variable to the input_vars variable as a vector value.  
> output_vars = c(colnames(output))  
    - Assign the output variable to the output_vars variable as a vector  
      value.  
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',  
  paste(input_vars, collapse = '+'))):  
    - Assign the function expression of the Naive Bayes model to the form  
      variable using the paste function.  
> form: Dispaly the function expression of the Naive Bayes model.  
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))  
    - Sample tdata in a 5 to 5 ratio.  
> tr_data=tdata[ind==1,]  
    - Assign the first sample (50%) to the training data (tr_data).  
> te_data=tdata[ind==2,]  
    - Assign the second sample (50%) to test data (te_data).  
> train_data.lda=naiveBayes(form,data=tr_data)  
    - Create a model function (classifier) by executing the Naive Bayes  
      Classification model with the tr_data data set.  
> ldapred=predict(train_data.lda, te_data, type='class')  
    - Generate a classification group by performing model prediction with  
      the te_data data set using the classifier (train_data.lda).  
> table(te_data$Attitude, ldapred)  
    - Execute model evaluation of the practical group and the prediction  
      group for model comparison.
```



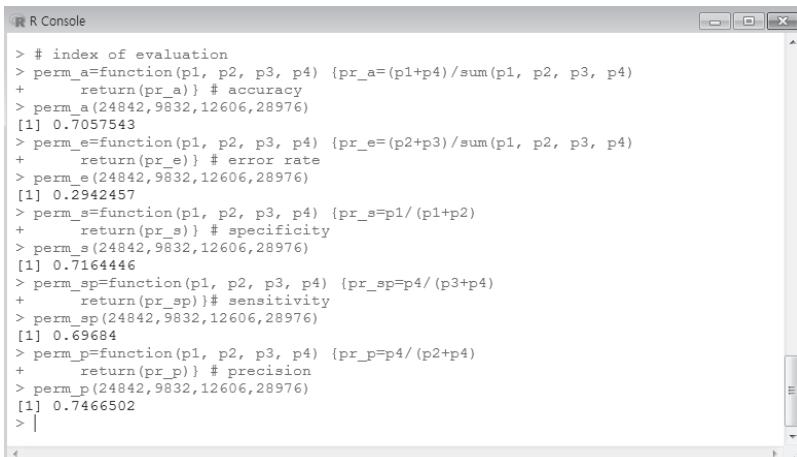
```

R R Console
> #1 naiveBayes classification model(atitude)
>
> rml(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('MASS')

There is a binary version available but the source version is later:
  binary source needs_compilation
MASS 7.3-49 7.3-50          TRUE

Binaries will be installed
Warning: package 'MASS' is in use and will not be installed
> library(MASS)
> install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
> library(e1071)
>
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning in read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> train_data.lda=naiveBayes(form,data=tr_data)
> ldpred=predict(train_data.lda, te_data, type='class')
> table(te_data$Attitude, ldpred)
      ldpred
Negative Positive
Negative     24745     9996
Positive     12423    29200
> |

```



```

R R Console
> # index of evaluation
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(24842,9832,12606,28976)
[1] 0.7057543
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(24842,9832,12606,28976)
[1] 0.2942457
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(24842,9832,12606,28976)
[1] 0.7164446
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)}# sensitivity
> perm_sp(24842,9832,12606,28976)
[1] 0.69684
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(24842,9832,12606,28976)
[1] 0.7466502
> |

```

■ Cyber bullying Type Prediction Model Evaluation

```
R Console

> library(MASS)
> install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
> library(e1071)
>
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning in read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> train_data.lda=naiveBayes(form,data=tr_data)
>
> ldapred=predict(train_data.lda, te_data, type='class')
> table(te_data$type, ldapred)

ldapred
      Bystander Complex Perpetrator Victim
Bystander          0    199     392   2257
Complex             0    214     187   1883
Perpetrator          0    119     1656   1881
Victim              0   1283     3125  16817
> |
```

```
R Console

> # index of evaluation
>
> perm_a=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ (pr_a=(p1+p6+p14+p16)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_a)) # accuracy
> perm_a(0,199,392,2257,0,214,187,1883,0,119,1656,1881,0,1283,3125,16817)
[1] 0.6226301936
>
> perm_u=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ (pr_u=(p2+p3+p4+p7+p8+p12)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_u)) # Upward accuracy
> perm_u(0,199,392,2257,0,214,187,1883,0,119,1656,1881,0,1283,3125,16817)
[1] 0.2265351681
>
> perm_d=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ (pr_d=(p5+p9+p10+p13+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_d)) # Downward accuracy
> perm_d(0,199,392,2257,0,214,187,1883,0,119,1656,1881,0,1283,3125,16817)
[1] 0.1508346383
>
> perm_e=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ (pr_e=(p2+p3+p4+p7+p8+p12+p5+p9+p10+p13+p14+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_e)) # Error rate
> perm_e(0,199,392,2257,0,214,187,1883,0,119,1656,1881,0,1283,3125,16817)
[1] 0.3773698064
> |
```

Neural Network Model Evaluation

■ Cyber bullying Risk (Negative, Positive) Prediction Model Evaluation

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("nnet")
> library(nnet)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> tr.nnet = nnet(form, data=tr_data, size=5)
> names(tr.nnet)
> summary(tr.nnet)
> p=predict(tr.nnet, te_data, type='class')
> table(te_data$Attitude,p)
```

```

R Console

> library(MASS)
>
> tdata = read.table('cyberbullying_attitude_S.txt', header=T)
> input=read.table('input_GST.txt', header=T, sep=",")
Warning in read.table('input_GST.txt', header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table("output_attitude.txt", header=T, sep=",")
Warning in read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T, prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> tr.nnet = nnet(form, data=tr_data, size=5)
# weights:  51
initial value 59420.367826
iter  10 value 44405.468652
iter  20 value 44051.769457
iter  30 value 43666.130018
iter  40 value 43235.006607
iter  50 value 43092.537807
iter  60 value 43058.709185
iter  70 value 42996.097842
iter  80 value 42984.747849
iter  90 value 42977.666527
iter 100 value 42969.354314
final value 42969.354314
stopped after 100 iterations
> p=predict(tr.nnet, te_data, type='class')
> table(te_data$Attitude,p)

          p
        Negative Positive
Negative    22535     11977
Positive     9351     32101
> |

```

```

R Console

> # index of evaluation
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(22820,11917,9429,32317)
[1] 0.7209053
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(22820,11917,9429,32317)
[1] 0.2790947
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(22820,11917,9429,32317)
[1] 0.6569364
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)}# sensitivity
> perm_sp(22820,11917,9429,32317)
[1] 0.774134
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(22820,11917,9429,32317)
[1] 0.7305919
> |

```

■ Cyber bullying Type Prediction Model Evaluation

```
R Console
> library(MASS)
>
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning in read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> tr.nnet = nnet(form, data=tr_data, size=5)
# weights: 69
initial value 28080.289039
iter 10 value 26838.475855
iter 20 value 26281.157293
iter 30 value 26067.701695
iter 40 value 25899.969188
iter 50 value 25816.355336
iter 60 value 25762.608868
iter 70 value 25733.116905
iter 80 value 25680.824074
iter 90 value 25672.005451
iter 100 value 25668.758376
final value 25668.758376
stopped after 100 iterations
> p=predict(tr.nnet, te_data, type='class')
> table(te_data$type,p)

      p
      Complex Perpetrator Victim
Bystander     16        78   2663
Complex       61        56   2171
Perpetrator     9        823   2774
Victim        59        433   20740
> |
```

```
R Console
> # index of evaluation
>
> perm_a=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {(pr_a=(p1+p8+p11+p16)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_a)} # accuracy
> perm_a(0,16,78,2663,0,61,56,2171,0,9,823,2774,0,59,433,20740)
[1] 0.7236221263
>
> perm_u=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {(pr_u=(p2+p3+p4+p7+p8+p12)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_u)} # Upward accuracy
> perm_u(0,16,78,2663,0,61,56,2171,0,9,823,2774,0,59,433,20740)
[1] 0.2596124887
>
> perm_d=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {(pr_d=(p5+p9+p10+p13+p14+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_d)} # Downward accuracy
> perm_d(0,16,78,2663,0,61,56,2171,0,9,823,2774,0,59,433,20740)
[1] 0.016765385
>
> perm_e=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {(pr_e=(c2*p3+p4*p7+p8*p12+p5*p9+p10*p13+p14*p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_e)} # Error rate
> perm_e(0,16,78,2663,0,61,56,2171,0,9,823,2774,0,59,433,20740)
[1] 0.2763778737
> |
```

Logistic Regression Model Evaluation

■ Cyber bullying Risk (Negative, Positive) Prediction Model Evaluation

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
  - To evaluate the predictive model (model function) with the logistic
    regression model, the category of the dependent variables (Attitude)
    included in the training data is Numeric (Negative: 0, Positive: 1).
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
> form
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> i_logistic=glm(form, family=binomial,data=tr_data)
> p=predict(i_logistic,te_data,type='response')
> p=round(p): Round the prediction probability and save it in the p object.
> table(te_data$Attitude,p)
```

```
R Console
> #3 logistic regression model(attitude)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning in read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')) )
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> i_logistic=glm(form, family=binomial,data=tr_data)
> p=predict(i_logistic,te_data,type='response')
> p=round(p)
> table(te_data$Attitude,p)
  p
    0     1
  0 22890 11835
  1  9929 31597
> |
```

```
R Console
> # index of evaluation
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(22523,11854,9939,31606)
[1] 0.7129554
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(22523,11854,9939,31606)
[1] 0.2870446
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(22523,11854,9939,31606)
[1] 0.6551764
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)}# sensitivity
> perm_sp(22523,11854,9939,31606)
[1] 0.7607654
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(22523,11854,9939,31606)
[1] 0.7272434
> |
```

■ Cyber bullying Type Prediction Model Evaluation

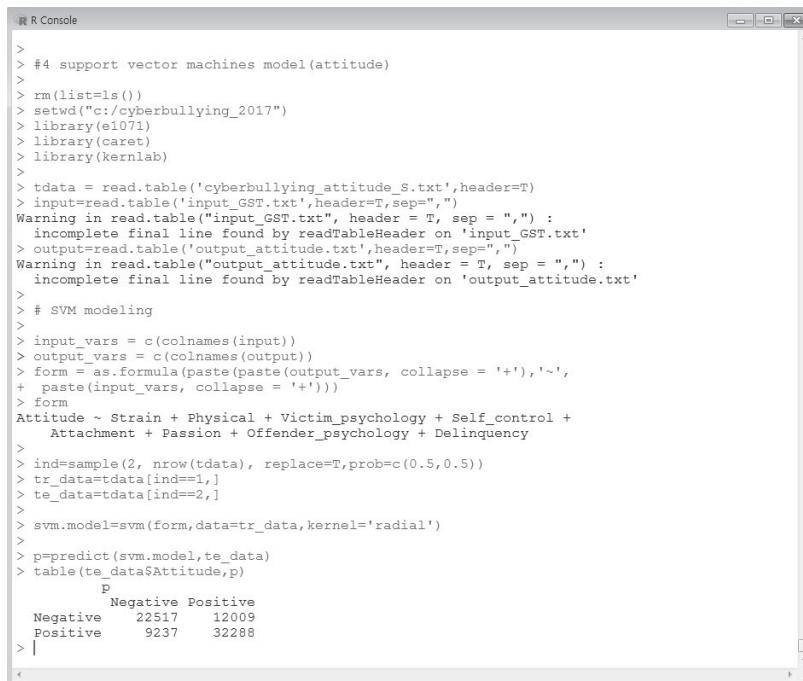
```
R Console
> library(MASS)
>
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning in read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
>
> # logistic modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> i_logistic=multinom(form, data=tr_data)
# weights:  40 (27 variable)
initial value 41441.883631
iter  10 value 26990.556964
iter  20 value 26566.658067
iter  30 value 26461.836100
final value 26460.025622
converged
> p=predict(i_logistic,te_data,type='class')
> table(te_data$Type,p)
      p
Bystander Complex Perpetrator Victim
Bystander      0      0       0 2792
Complex        0      0       0 2308
Perpetrator     0      0       0 3681
Victim         0      0       0 20997
> |
```

```
R Console
> # index of evaluation
>
> perm_a=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_a=(p1+p6+p11+p16)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+   return(pr_a)} # accuracy
> perm_a(0,0,0,2792,0,0,0,2308,0,0,0,3681,0,0,0,20997)
[1] 0.7051178723
>
> perm_u=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_u=(p2+p3+p4+p7+p8+p12)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+   return(pr_u)} # Upward accuracy
> perm_u(0,0,0,2792,0,0,0,2308,0,0,0,3681,0,0,0,20997)
[1] 0.2948821277
>
> perm_d=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_d=(p5+p9+p10+p13+p14+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+   return(pr_d)} # Downward accuracy
> perm_d(0,0,0,2792,0,0,0,2308,0,0,0,3681,0,0,0,20997)
[1] 0
>
> perm_e=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_e=(p2+p3+p4+p7+p8+p12+p5+p9+p10+p13+p14+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+   return(pr_e)} # Error rate
> perm_e(0,0,0,2792,0,0,0,2308,0,0,0,3681,0,0,0,20997)
[1] 0.2948821277
> |
```

Support Vector Machine Model Evaluation

■ Cyber bullying Risk (Negative, Positive) Prediction Model Evaluation

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> library(e1071)
> library(caret)
> library(kernlab)
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> svm.model=svm(form,data=tr_data,kernel='radial')
> p=predict(svm.model,te_data)
> table(te_data$Attitude,p)
```



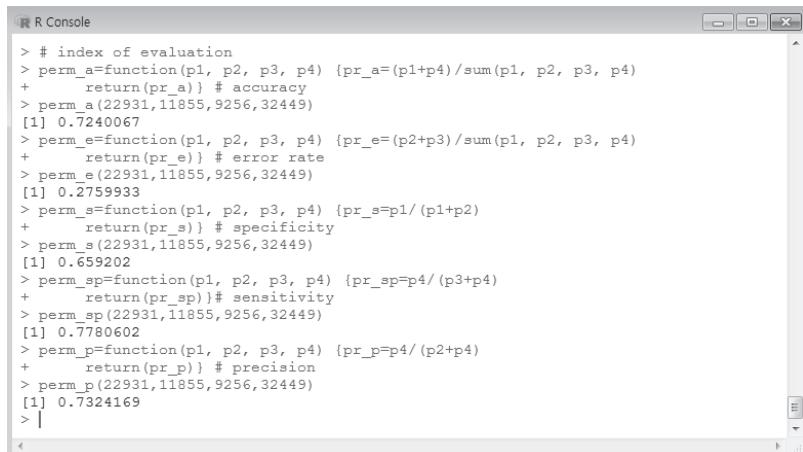
```

R Console

> #4 support vector machines model(attitude)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> library(e1071)
> library(caret)
> library(kernlab)
>
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep="")
Warning in read.table("input_GST.txt", header = T, sep = "") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep="")
Warning in read.table("output_attitude.txt", header = T, sep = "") :
  incomplete final line found by readTableHeader on 'output_attitude.txt',
> #
> # SVM modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T, prob=(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> svm.model=svm(form,data=tr_data,kernel='radial')
>
> p=predict(svm.model,te_data)
> table(te_data$Attitude,p)

          Negative Positive
Negative      22517    12009
Positive       9237    32288
> |

```



```

R Console

> # index of evaluation
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(22931,11855,9256,32449)
[1] 0.7240067
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(22931,11855,9256,32449)
[1] 0.2759933
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(22931,11855,9256,32449)
[1] 0.659202
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)}# sensitivity
> perm_sp(22931,11855,9256,32449)
[1] 0.7780602
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(22931,11855,9256,32449)
[1] 0.7324169
> |

```

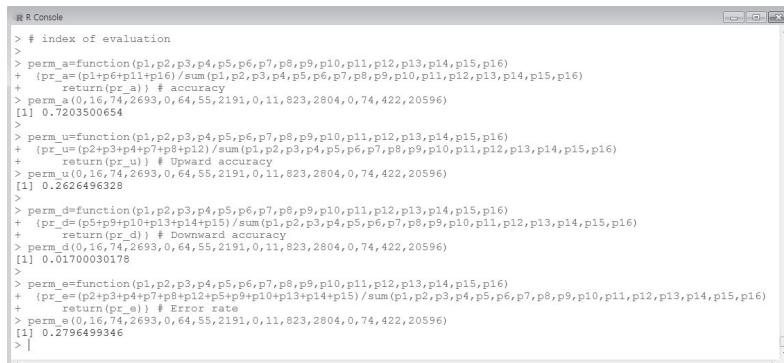
■ Cyber bullying Type Prediction Model Evaluation



```

R R Console
> # 4.1 support vector machines model(type)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
> library(e1071)
> install.packages('caret')
Warning: package 'caret' is in use and will not be installed
> library(caret)
> install.packages('kernlab')
Warning: package 'kernlab' is in use and will not be installed
> library(kernlab)
>
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning in read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt',
>
> # support vector modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> svm.model=svm(form,data=tr_data,kernel='radial')
>
> p=predict(svm.model,te_data)
> table(te_data$Type,p)
      P
      Bystander Complex Perpetrator Victim
Bystander          0     16        74   2693
Complex            0      64        55   2191
Perpetrator         0      11       823   2804
Victim             0      74       422   20596
> |

```



```

R R Console
> # index of evaluation
>
> perm_a=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_a=(p1+p6*p11+p16)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_a)} # accuracy
> perm_a(0,16,74,2693,0,64,55,2191,0,11,823,2804,0,74,422,20596)
[1] 0.7203500654
>
> perm_u=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_u=(p2+p3*p4+p7*p8*p12)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_u)} # Upward accuracy
> perm_u(0,16,74,2693,0,64,55,2191,0,11,823,2804,0,74,422,20596)
[1] 0.2626496328
>
> perm_d=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_d=(p5*p9*p10*p13*p14*p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_d)} # Downward accuracy
> perm_d(0,16,74,2693,0,64,55,2191,0,11,823,2804,0,74,422,20596)
[1] 0.01700030178
>
> perm_e=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_e=(p2*p3*p4*p7*p8*p12*p5*p9*p10*p13*p14*p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ + return(pr_e)} # Error rate
> perm_e(0,16,74,2693,0,64,55,2191,0,11,823,2804,0,74,422,20596)
[1] 0.2796499346
> |

```

Random Forest Model Evaluation

■ Cyber bullying Risk (Negative, Positive) Prediction Model Evaluation

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("randomForest")
> library(randomForest)
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> tdata.rf = randomForest(form, data=tr_data,
  forest=FALSE,importance=TRUE)
> p=predict(tdata.rf,te_data)
> table(te_data$Attitude,p)
```

```
R Console

> 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

>
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning in read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> # random forests modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> indi=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[indi==1,]
> te_data=tdata[indi==2,]
>
> tdata.rf = randomForest(form, data=tr_data ,forest=FALSE,importance=TRUE)
>
> p=predict(tdata.rf,te_data)
> table(te_data$Attitude,p)
   p
Negative Positive
Negative    21790    12854
Positive     8317    32974
> |
```

```
R Console

> # index of evaluation
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(21998,12574,8736,33139)
[1] 0.7212448
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(21998,12574,8736,33139)
[1] 0.2787552
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(21998,12574,8736,33139)
[1] 0.6362953
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)}# sensitivity
> perm_sp(21998,12574,8736,33139)
[1] 0.7913791
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(21998,12574,8736,33139)
[1] 0.724936
> |
```

■ Cyber bullying Type Prediction Model Evaluation

```

R Console

> # 5.1 random forests model(type)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("randomForest")
Warning: package 'randomForest' is in use and will not be installed
> library(randomForest)
>
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table("output_type.txt",header=T,sep=",")
Warning in read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
>
> # random forests modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> tdata.rf = randomForest(form, data=tr_data ,forest=FALSE,importance=TRUE)
>
> p=predict(tdata.rf,te_data)
> table(te_data$type,p)

      P
      Bystander Complex Perpetrator Victim
Bystander          0     0        0   2793
Complex            0     0        0   2290
Perpetrator         0     0        0   3547
Victim             0     0        0  20998
> |

```

```

R Console

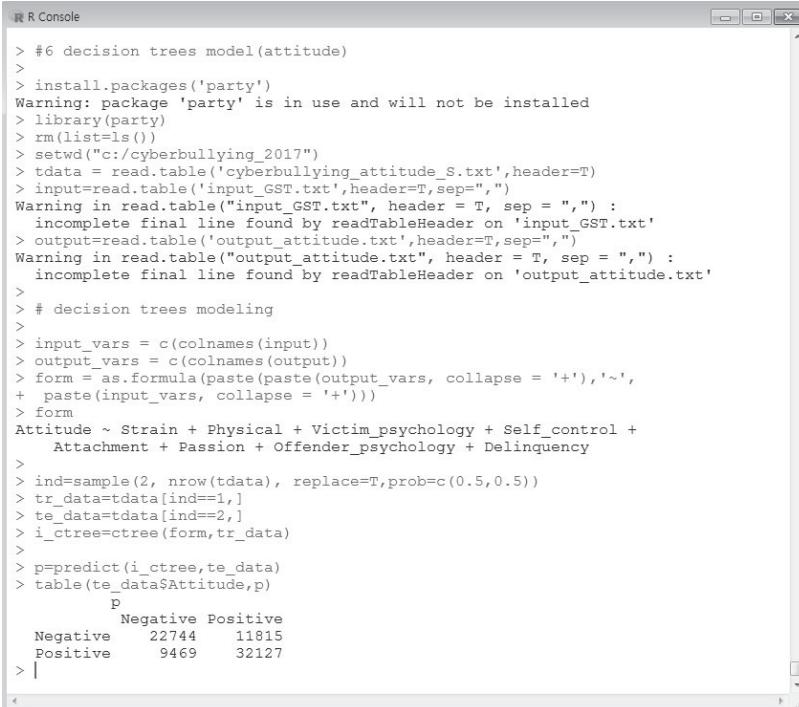
> # index of evaluation
>
> perm_u=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_a=(p1+p6+p11+p16)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+   return(pr_a)} # accuracy
> perm_a(0,0,0,2793,0,0,0,2290,0,0,0,3547,0,0,0,20998)
[1] 0.7087214797
>
> perm_u=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_u=(p2+p3+p4+p7+p12)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+   return(pr_u)} # Upward accuracy
> perm_u(0,0,0,2793,0,0,0,2290,0,0,0,3547,0,0,0,20998)
[1] 0.2912785203
>
> perm_d=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_d=(p5+p9+p10+p13+p14+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+   return(pr_d)} # Downward accuracy
> perm_d(0,0,0,2793,0,0,0,2290,0,0,0,3547,0,0,0,20998)
[1] 0
>
> perm_e=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_e=(p2+p3+p4+p7+p8+p12+p5+p9+p10+p13+p14+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+   return(pr_e)} # Error rate
> perm_e(0,0,0,2793,0,0,0,2290,0,0,0,3547,0,0,0,20998)
[1] 0.2912785203
> |

```

Decision Tree Model Evaluation

■ Cyber bullying Risk (Negative, Positive) Prediction Model Evaluation

```
> install.packages('party')
> library(party)
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> i_ctree=ctree(form,tr_data)
> p=predict(i_ctree,te_data)
> table(te_data$Attitude,p)
```

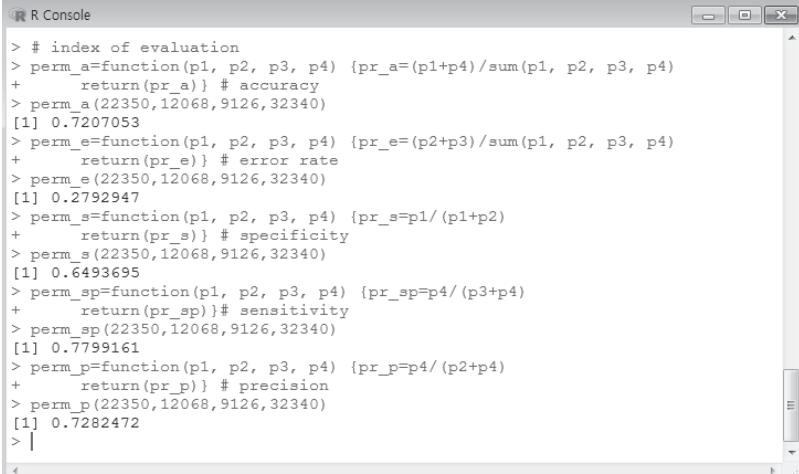


```

R R Console

> # 6 decision trees model(attitude)
>
> install.packages('party')
Warning: package 'party' is in use and will not be installed
> library(party)
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning in read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> # decision trees modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> i_ctree=ctree(form,tr_data)
>
> p=predict(i_ctree,te_data)
> table(te_data$Attitude,p)
      p
    Negative Positive
Negative     22744     11815
Positive      9469     32127
> |

```



```

R R Console

> # index of evaluation
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(22350,12068,9126,32340)
[1] 0.7207053
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(22350,12068,9126,32340)
[1] 0.2792947
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(22350,12068,9126,32340)
[1] 0.6493695
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)}# sensitivity
> perm_sp(22350,12068,9126,32340)
[1] 0.7799161
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(22350,12068,9126,32340)
[1] 0.7282472
> |

```

■ Cyber bullying Type Prediction Model Evaluation

```
R Console
> # 6.1 decision trees model(type)
>
> install.packages('party')
Warning: package 'party' is in use and will not be installed
> library(party)
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning in read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
>
> # decision trees modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> i_ctree=ctree(form,tr_data)
>
> p=predict(i_ctree,te_data)
> table(te_data$Type,p)

      Bystander Complex Perpetrator Victim
Bystander          4      1       66   2696
Complex            1      8       57   2267
Perpetrator         1      0       817  2788
Victim             6      8      400  20718
> |
```

```
R Console
> # index of evaluation
>
> perm_a=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_a=(p1+p6+p11+p16)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ return(pr_a)} # accuracy
> perm_a(4,1,66,2696,1,8,57,2267,1,0,817,2788,6,8,400,20718)
[1] 0.7221328507
>
> perm_u=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_u=(p2+p3+p4+p7+p8+p12)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ return(pr_u)} # Upward accuracy
> perm_u(4,1,66,2696,1,8,57,2267,1,0,817,2788,6,8,400,20718)
[1] 0.2639251961
>
> perm_d=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_d=(p5+p9+p10+p13+p14+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ return(pr_d)} # Downward accuracy
> perm_d(4,1,66,2696,1,8,57,2267,1,0,817,2788,6,8,400,20718)
[1] 0.01394195321
>
> perm_e=function(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ {pr_e=(p2+p3+p4+p7+p8+p12+p5+p9+p10+p13+p14+p15)/sum(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16)
+ return(pr_e)} # Error rate
> perm_e(4,1,66,2696,1,8,57,2267,1,0,817,2788,6,8,400,20718)
[1] 0.2778671493
> |
```

Machine Learning Model Evaluation Using ROC Curves

Below, ROC curves are used to evaluate the machine learning models that predict cyber bullying risk and types.

NaïveBayes ROC

```
> rm(list=ls()): Initialize all variables.
> setwd("c:/cyberbullying_2017"): Set the working directory.
> install.packages('MASS'): Install the 'MASS' package.
> library(MASS)
  - Load the MASS package containing the write.matrix () function.
> install.packages('e1071'): Install the e1071 package.
> library(e1071): Load the e1071 package.
> install.packages('ROCR')
  - Install the package that generates the ROC curve.
> library(ROCR): Load the ROCR package.
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
  - Assign the cyber bullying data file to the tdata object.
.> input=read.table('input_GST.txt',header=T,sep=",")
  - Assign the independent variable to the input object as a delimiter (,).
> output=read.table('output_attitude.txt',header=T,sep=",")
  - Assign the dependent variable to the output object as a delimiter (,).
> p_output=read.table('p_output_bayes.txt',header=T,sep=",")
  - Assign the predictive random variable (posterior.0, posterior.1) to the
    p_output object as a delimiter (,).
> input_vars = c(colnames(input))
  - Assign the input variable to the input_vars variable as a vector value.
> output_vars = c(colnames(output))
  - Assign the p_output variable to the p_output_vars variable as a vector
    value.
> p_output_vars = c(colnames(p_output))
  - Assign the p_output variable to the p_output_vars variable as a vector
    value.
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
  - Assign the function expression of the Naive Bayes model to a form
    variable using a function that combines strings (paste).
> form: Display the function expression of the Naive Bayes model.
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
  - Sample tdata in a 5 to 5 ratio.
```

```
> tr_data=tdata[ind==1,]
  - Assign the first sample (50%) to training data (tr_data).
> te_data=tdata[ind==2,]
  - Assign the second sample (50%) to the test data (te_data).
> train_data.lda=naiveBayes(form,data=tr_data)
  - Create a model function (classifier) by executing the Naive Bayes Classification model with the tr_data data set.
  - train_data.lda=naiveBayes(form,data=tr_data, laplace=1)
> p=predict(train_data.lda, te_data, type='raw')
  - Generate classification group by executing model prediction with the test_data set by using classifier (train_data.lda).
> dimnames(p)=list(NULL,c(p_output_vars))
  - Assign the probability values of the predicted dependent variables to posterior.0 (positive predictive probability) and posterior.1 (negative predictive probability) variables.
> summary(p)
> mydata=cbind(te_data, p)
  - Append the posterior.0 and posterior.1 variables to the te_data dataset and assign it to the mydata object.
> write.matrix(mydata,'naive_bayse_cyberbullying_ROC.txt')
  - Save the mydata object as 'naive_bayse_cyberbullying_ROC.txt' file.
> mydata1=read.table('naive_bayse_cyberbullying_ROC.txt',header=T)
  - Assign the naive_bayse_cyberbullying_ROC.txt file to the mydata1 object.
> attach(mydata1)
> pr=prediction(posterior.1, te_data$Attitude)
  - Estimate the estimation of tdata's Attitude using real group and predictive group.
> bayes_prf=performance(pr, measure='tpr', x.measure='fpr')
  - Assign the true positive rate (tpr) and false positive rate (fpr) of the ROC curve to the bayes_prf object.
  - TPR: sensitivity, FPR: 1-specificity
> auc=performance(pr, measure='auc')
  - Evaluate the performance of the AUC curve.
> auc_bayes=auc@y.values[[1]]
  - Calculate the AUC statistic and assign it to the auc_bayes object.
> auc_bayes: Dispaly AUC statistics on the screen.
  - auc_bayes=sprintf("%.2f",auc_bayes): Display the numbers with tenth and hundredth decimals.
> plot(bayes_prf,col=1,lty=1,lwd=1.5,main='ROC curver for Machine Learning Models')
```

- Draw ROC curves with a title, 'ROC curves for Machine Learning Models'.

> abline(0,1,lty=3): Draw the baseline of the ROC curve.

Neural Networks ROC

```

> install.packages("nnet")
> library(nnet)
> attach(tdata)
> tr.nnet = nnet(form, data=tr_data, size=5)
> p=predict(tr.nnet, te_data, type='raw')
> pr=prediction(p, te_data$Attitude)
> neural_prf=performance(pr, measure='tpr', x.measure='fpr')
> neural_x=unlist(attr(neural_prf, 'x.values'))
    - Assign the value of the x-axis (fpr) to the neural_x object.
> neural_y=unlist(attr(neural_prf, 'y.values'))
    - Assign the y-axis value (tpr) to the neural_y object.
> auc=performance(pr, measure='auc')
> auc_neural=auc@y.values[[1]]
> auc_neural
    - auc_neural=sprintf("%.2f",auc_neural): Display the numbers with
      tenth and hundredth decimals.
> lines(neural_x,neural_y, col=2,lty=2)
    - Display fpr on the screen in the form of red (col = 2) and dashed line
      (lty = 2) with the values of the X axis and tpr as Y values.

```

Logistic ROC

```

> i_logistic=glm(form, family=binomial,data=tr_data)
> p=predict(i_logistic,te_data,type='response')
> pr=prediction(p, te_data$Attitude)
> lo_prf=performance(pr, measure='tpr', x.measure='fpr')
> lo_x=unlist(attr(lo_prf, 'x.values'))
> lo_y=unlist(attr(lo_prf, 'y.values'))
> auc=performance(pr, measure='auc')
> auc_lo=auc@y.values[[1]]
> auc_lo
    - auc_lo=sprintf("%.2f",auc_lo): Display the numbers with tenth and
      hundredth decimals.
> lines(lo_x,lo_y, col=3,lty=3)
    - Display in green color (col=3) and dotted lines (lty=3) .

```

SVM ROC

```
> library(e1071)
> library(caret)
> install.packages('kernlab')
> library(kernlab)
> svm.model=svm(form,data=tr_data,kernel='radial')
> p=predict(svm.model,te_data)
> pr=prediction(p, te_data$Attitude)
> svm_prf=performance(pr, measure='tpr', x.measure='fpr')
> svm_x=unlist(attr(svm_prf, 'x.values'))
> svm_y=unlist(attr(svm_prf, 'y.values'))
> auc=performance(pr, measure='auc')
> auc_svm=auc@y.values[[1]]
> auc_svm
- auc_svm=sprintf("%.2f",auc_svm): Display the numbers with tenth
and hundredth decimals.
> lines(svm_x,svm_y, col=4,lty=4)
- Display on the screen in blue color (col=4) and dashed lines (ity=4) .
```

Random Forests ROC

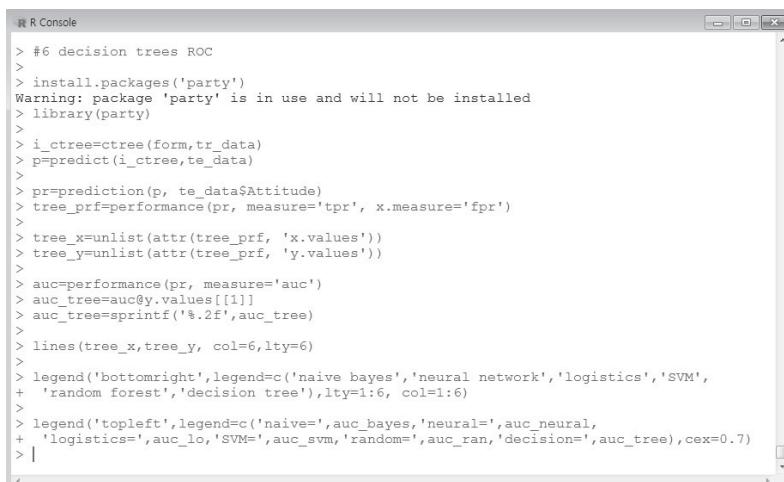
```
> install.packages("randomForest")
> library(randomForest)
> tdata.rf = randomForest(form, data=tr_data ,forest=FALSE,
  importance=TRUE)
> p=predict(tdata.rf,te_data)
> pr=prediction(p, te_data$Attitude)
> ran_prf=performance(pr, measure='tpr', x.measure='fpr')
> ran_x=unlist(attr(ran_prf, 'x.values'))
> ran_y=unlist(attr(ran_prf, 'y.values'))
> auc=performance(pr, measure='auc')
> auc_ran=auc@y.values[[1]]
> auc_ran
- auc_ran=sprintf("%.2f",auc_ran): Display the numbers with tenth and
hundredth decimals.
> lines(ran_x,ran_y, col=5,lty=5)
- Dispaly on the screen in light blue color (col=5) and long dashed
lines (ity=5) .
```

Decision Trees ROC

```
> install.packages('party')
> library(party)
> i_ctree=ctree(form,tr_data)
> p=predict(i_ctree,te_data)
> pr=prediction(p, te_data$Attitude)
> tree_prf=performance(pr, measure='tpr', x.measure='fpr')
> tree_x=unlist(attr(tree_prf, 'x.values'))
> tree_y=unlist(attr(tree_prf, 'y.values'))
> auc=performance(pr, measure='auc')
> auc_tree=auc@y.values[[1]]
> auc_tree
- auc_tree=sprintf("%.2f",auc_tree): Display the numbers with tenth and
  hundredth decimals.
> lines(tree_x,tree_y, col=6,lty=5)
- Display on the screen in purple color (col=5) and two long dashes
  (lty=6).
> legend('bottomright',legend=c('naive bayes','neural network',
  'logistics','SVM', 'random forest','decision tree'),lty=1:6, col=1:6)
- Set the legend of the machine learning model at bottomright.
> legend('topleft',legend=c('naive=',auc_bayes,'neural=',auc_neural,
  'logistics=',auc_lo,'SVM=',auc_svm,'random=',auc_ran,'decision=',
  auc_tree),cex=0.7)
- Set the legend of the AUC statistic for the machine learning model in
  topleft.
```



```
> library(ROCR)
>
> tdata = read.table('cyberbullying_attitude_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table('input_GST.txt', header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning in read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
> p_output=read.table('p_output_bayes.txt',header=T,sep=",")
Warning in read.table("p_output_bayes.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_bayes.txt',
  collapse = '+'), '~~',
+ paste(input_vars, collapse = '+'))"
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> train_data.lda=naiveBayes(form,data=tr_data)
>
> p=predict(train_data.lda, te_data, type='raw')
>
> dimnames(p)=list(NULL,c(p_output_vars))
> #summary(p)
>
> mydata=cbind(te_data, p)
> write.matrix(mydata,'naive_bayse_cyberbullying_ROC.txt')
>
> mydata=read.table('naive_bayse_cyberbullying_ROC.txt',header=T)
> #attach(mydata)
> pr=prediction(mydata$posterior.1, te_data$Attitude)
> bayes_prf=performance(pr, measure='tpr', x.measure='fpr')
>
> auc=performance(pr, measure='auc')
> auc_bayes=auc@y.values[[1]]
> auc_bayes=sprintf('.2f',auc_bayes)
> plot(bayes_prf,col=1,lty=1,lwd=1.5,main='ROC curver for Machine Learning Models')
>
> abline(0,1,lty=3)
> |
```



```
> #6 decision trees ROC
>
> install.packages('party')
Warning: package 'party' is in use and will not be installed
> library(party)
>
> i_ctree=ctree(form,tr_data)
> p=predict(i_ctree,te_data)
>
> pr=prediction(p, te_data$Attitude)
> tree_prf=performance(pr, measure='tpr', x.measure='fpr')
>
> tree_x=unlist(attr(tree_prf, 'x.values'))
> tree_y=unlist(attr(tree_prf, 'y.values'))
>
> auc=performance(pr, measure='auc')
> auc_tree=auc@y.values[[1]]
> auc_tree=sprintf('.2f',auc_tree)
>
> lines(tree_x,tree_y, col=6,lty=6)
>
> legend('bottomright',legend=c('naive bayes','neural network','logistics','SVM',
+ 'random forest','decision tree'),lty=1:6, col=1:6)
>
> legend('topleft',legend=c('naive=',auc_bayes,'neural=',auc_neural,
+ 'logistics=',auc_lo,'SVM=',auc_svm,'random=',auc_ran,'decision=',auc_tree),cex=0.7)
> |
```

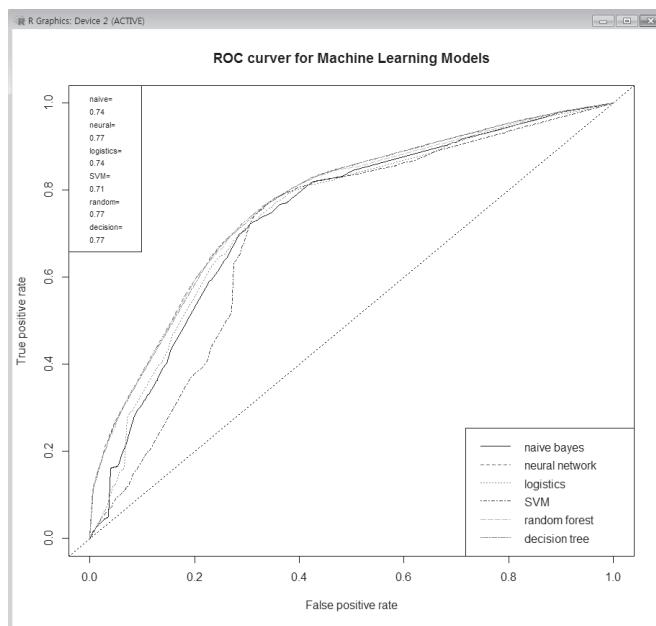


Fig. 11. The receiver operator characteristic curve for Machine learning models

Table 5 Evaluation of machine learning models

Evaluation Index	Naïve Bayes classification	neural networks	logistic regression	support vector machines	random forests	decision trees
accuracy	70.58	72.09	71.30	72.40	72.12	72.07
error rate	29.42	27.91	28.70	27.60	27.88	27.93
specificity	71.64	65.69	65.52	65.92	63.63	64.94
sensitivity	69.68	77.41	76.08	77.81	79.14	77.99
precision	74.67	73.06	72.72	73.42	72.49	72.82
AUC	0.74	0.77	0.74	0.71	0.77	0.77
best accuracy				support vector machines		
best error rate				support vector machines		
best specificity				Naïve Bayes classification		
best sensitivity				random		
best precision				Naïve Bayes classification		
best AUC(Area Under the Curve)				neural, random, trees		

ARTIFICIAL INTELLIGENCE

Calculate the Effect of Input Variables on Output Variables (Prediction Probability)

- Cyber bullying Type Prediction Model Development (Neural Network Prediction Model)

```
> # neural networks modeling nnet(type)
> # Utilizing Artificial Intelligence (Estimation of Probability Values)
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("nnet")
> library(nnet)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_type.txt',header=T,sep=",")
> p_output=read.table('p_output_type_n.txt',header=T,sep=",")
> attach(tdata)
> ID=cbind(Year,Month,Day,Hour,Strain,Physical,Victim_psychology,
  Self_control,Attachment,Passion,Offender_psychology,Delinquency)
  - In the learning data, identification (Year, Month, Day, Hour) and
    input variables are selected and assigned to the ID object.
> # neural networks modeling
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
> paste(input_vars, collapse = '+'))))
> form
> p_output_vars = c(colnames(p_output))
> tr.nnet = nnet(form, data=tdata, size=5)
> p=predict(tr.nnet, tdata, type='raw')
> dimnames(p)=list(NULL,c(p_output_vars))
> pred_obs = cbind(ID, p)
  - Append the random variable of cyber violence type to the ID data set
```

and then assign to the pred_obs object.

```
> write.matrix(pred_obs,'cyberbullying_type_neural_prob.txt')
  - Save the pred_obs object as 'cyberbullying_type_neural_prob.txt' file.
> mydata1=read.table('cyberbullying_type_neural_prob.txt',header=T)
> attach(mydata1)
> mean(mydata1$p_Perpetrator)
  - Display the predictive probability value of the offender.
> mean(mydata1$p_Victim)
  - Display the predicted probability value of the victim.
> mean(mydata1$p_Bystander)
  - Display the predictive probability value of the bystander.
> mean(mydata1$p_Complex)
  - Display the predictive probability value of the complex.
```

The screenshot shows the R Console window with the following R script:

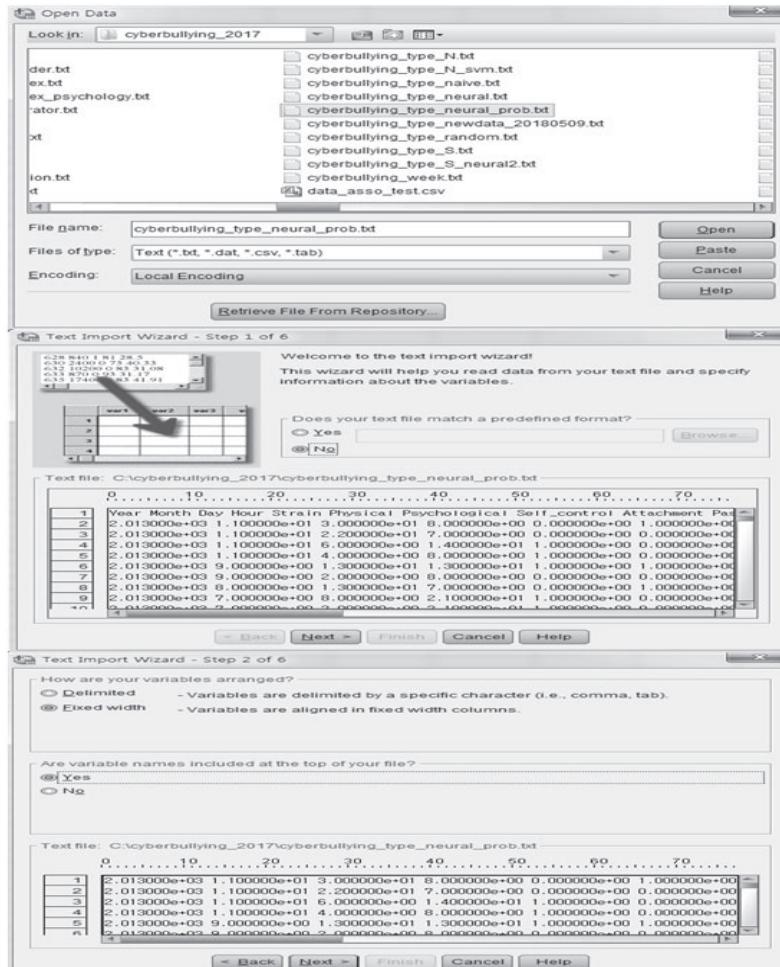
```
> ID=cbind(Year,Month,Day,Hour,Strain,Physical,Victim_psychology,Self_control,
+ Attachment,Passion,Offender_psychology,Delinquency)
>
> # neural networks modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
> p_output_vars = c(colnames(p_output))
>
> tr.nnet = nnet(form, data=tdata, size=5)
# weights: 69
initial value 86628.826373
iter 10 value 53085.978513
iter 20 value 52133.086768
iter 30 value 51986.730072
iter 40 value 51836.789813
iter 50 value 51641.690925
iter 60 value 51468.534602
iter 70 value 51398.520187
iter 80 value 51334.849972
iter 90 value 51299.094848
iter 100 value 51289.659538
final value 51289.659538
stopped after 100 iterations
> p=predict(tr.nnet, tdata, type='raw')
> dimnames(p)=list(NULL,c(p_output_vars))
> pred_obs = cbind(ID, p)
> write.matrix(pred_obs,'cyberbullying_type_neural_prob.txt')
> mydata1=read.table('cyberbullying_type_neural_prob.txt',header=T)
> #attach(mydata1)
> mean(mydata1$p_Perpetrator)
[1] 0.1218926321
> mean(mydata1$p_Victim)
[1] 0.7078061569
> mean(mydata1$p_Bystander)
[1] 0.0934012408
> mean(mydata1$p_Complex)
[1] 0.07689997019
> |
```

■ Cyber bullying neural-network prediction model's data conversion

The 'cyber bullying type_neural_prob.txt' file, which contains the predicted values for each type, is converted into an SPSS (statistical

software package) data file to analyze the effect of each input variable on the cyber bullying types.

Step 1: Use the [File – Open – Data] command to convert to an SPSS file.



Text Import Wizard - Fixed Width Step 3 of 6

The first case of data begins on which line number?

How many lines represent a case?

How many cases do you want to import?

All of the cases

The first cases.

A percentage of the cases: %

Data preview

	0	10	20	30	40	50	60	70
1	2.013000e+03	1.100000e-01	3.000000e+01	8.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
2	2.013000e+03	1.100000e-01	2.200000e-01	7.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
3	2.013000e+03	1.100000e-01	6.000000e-01	1.400000e+01	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
4	2.013000e+03	1.100000e-01	4.000000e-01	8.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
5	2.013000e+03	9.000000e-01	1.300000e+01	1.300000e+01	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
6	2.013000e+03	9.000000e-01	2.000000e+00	8.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
7	2.013000e+03	8.000000e-01	1.300000e+01	7.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
8	2.013000e+03	7.000000e-01	8.000000e+00	2.100000e+01	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
9	2.013000e+03	7.000000e-01	2.000000e+00	2.100000e+01	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
10	2.013000e+03	6.000000e-01	1.400000e-01	7.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
11	2.013000e+03	5.000000e-01	2.400000e+01	1.000000e+01	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
12	2.013000e+03	4.000000e-01	2.500000e+01	9.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00

< Back Next > Finish Cancel Help

Text Import Wizard - Fixed Width Step 4 of 6

Specify where each variable begins. The first column is column 0.

To INSERT a variable break line, click at the desired position in the ruler or data area. Alternatively, move to the position using the arrow keys or by typing the column number; then press the Insert Break button.

To MOVE a variable break line, drag it to the new position.

To DELETE a variable break line, select it or type its position. Then press the Delete key or the Delete Break button.

	0	10	20	30	40	50	60	70
1	2.013000e+03	1.100000e+01	3.000000e+01	8.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
2	2.013000e+03	1.100000e+01	2.200000e+01	7.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
3	2.013000e+03	1.100000e+01	6.000000e+01	1.400000e+01	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
4	2.013000e+03	1.100000e+01	4.000000e+01	8.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
5	2.013000e+03	9.000000e+00	1.300000e+01	1.300000e+01	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
6	2.013000e+03	9.000000e+00	2.000000e+01	8.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
7	2.013000e+03	8.000000e+00	1.300000e+01	7.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00
8	2.013000e+03	7.000000e+00	6.000000e+01	8.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00

Ruler: 0 10 20 30 40 50 60 70

Column Number: Insert Break Delete Break

Current Variable Width: 0

< Back Next > Finish Cancel Help

Text Import Wizard - Step 6 of 6

You have successfully defined the format of your text file.

Would you like to save this file format for future use?

Yes

No

Would you like to paste the syntax?

Yes Cache data locally

No

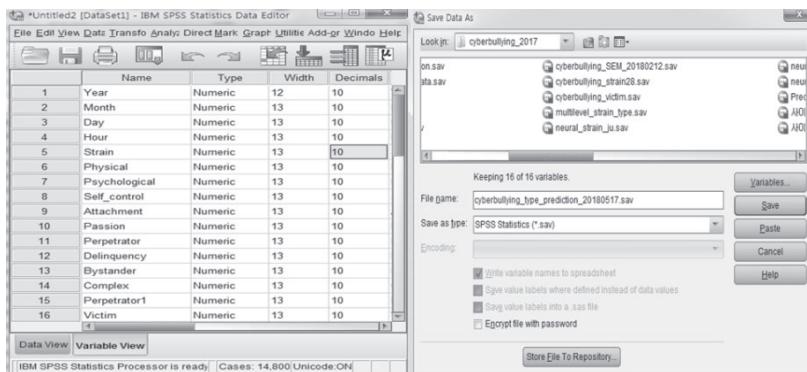
Press the Finish button to complete the text import wizard.

Data preview

YearMonthID	ayHourStrat	nPPhysicalS	ychologicalS	elf_controlA	ttachment	sionPerP
2.013000e-01	1.100000e-01	3.000000e-01	8.000000e-01	0.000000e+00	1.000000e+00	0.000000e+00
2.013000e-01	1.100000e-01	2.200000e-01	7.000000e-01	0.000000e+00	0.000000e+00	0.000000e+00
2.013000e-01	1.100000e-01	6.000000e-01	1.400000e-01	1.000000e+00	0.000000e+00	0.000000e+00
2.013000e-01	1.100000e-01	4.000000e-01	8.000000e-01	1.000000e+00	0.000000e+00	0.000000e+00
2.013000e-01	9.000000e-01	1.300000e-01	1.300000e-01	1.000000e+00	1.000000e+00	0.000000e+00
2.013000e-01	9.000000e-01	2.000000e-01	8.000000e-01	0.000000e+00	0.000000e+00	0.000000e+00
2.013000e-01	8.000000e-01	1.000000e-01	7.000000e-01	1.000000e+00	0.000000e+00	0.000000e+00
2.013000e-01	7.000000e-01	2.000000e-01	2.100000e-01	1.000000e+00	0.000000e+00	0.000000e+00
2.013000e-01	6.000000e-01	1.400000e-01	7.000000e-01	1.000000e+00	0.000000e+00	0.000000e+00
2.013000e-01	5.000000e-01	2.400000e-01	1.000000e-01	1.000000e+00	1.000000e+00	0.000000e+00
2.013000e-01	4.000000e-01	3.000000e-01	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00

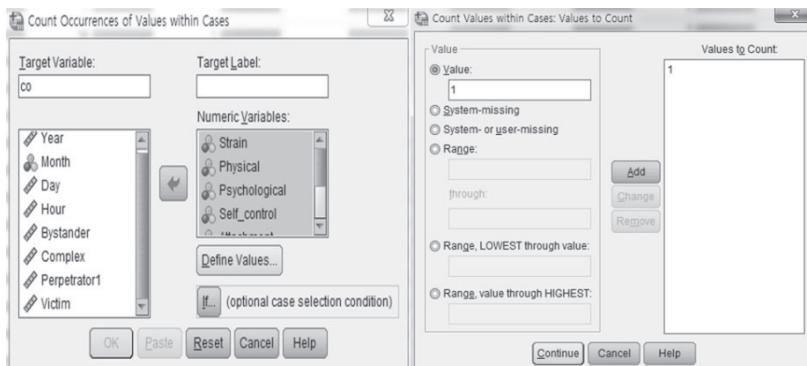
< Back Next > Finish Cancel Help

Step 2: Save the SPSS data file:
 cyberbullying_type_prediction_20180517.sav .



Step 3: Use the [Count – Occurrences of Values within Cases] command to create a count variable for the eight input variables.

- The count variable (co) stores the number of the eight input variables that simultaneously appeared in the document (coded as one for each input factor present). For example, if co = 2, it means that two of the eight input variables appeared simultaneously in the document.



The effect that the input variables have on the predictions for each cyber bullying type can be calculated using the following equation:

- Prediction probability of input factor = (Whether or not the input factor appears) × (Prediction probability of all input factors by type) / (Frequency of eight input factors appearing simultaneously in each document (i.e., the co)).
- Example: Compute B_Strain = Strain * Bystander / co.
where:
 - B_Strain: Strain factor's prediction probability.
 - Strain: Whether or not the Strain factor appears in the document (1 or 0).
 - Bystander, Complex, Perpetrator, Victim: Prediction probability of all input factors, for the bystander, complex, perpetrator, and victim types, respectively.
 - co: Frequency of eight input factors appearing simultaneously.

```

1 Encoding: UTF-8
2 compute B_Strain=Strain*Bystander/co.
3 compute B_Physical=Physical*Bystander/co.
4 compute B_Psychological=Psychological*Bystander/co.
5 compute B_Self_control=Self_control*Bystander/co.
6 compute B_Attachment=Attachment*Bystander/co.
7 compute B_Passion=Passion*Bystander/co.
8 compute B_Perpetrator=Perpetrator*Bystander/co.
9 compute B_Delinquency=Delinquency*Bystander/co.
10 compute B_pred=Bystander.
11 execute.
12 compute C_Strain=Strain*Complex/co.
13 compute C_Physical=Physical*Complex/co.
14 compute C_Psychological=Psychological*Complex/co.
15 compute C_Self_control=Self_control*Complex/co.
16 compute C_Attachment=Attachment*Complex/co.
17 compute C_Passion=Passion*Complex/co.
18 compute C_Perpetrator=Perpetrator*Complex/co.
19 compute C_Delinquency=Delinquency*Complex/co.
20 compute C_pred=Complex.
21 execute.
22 compute P_Strain=Strain*Perpetrator1/co.
23 compute P_Physical=Physical*Perpetrator1/co.
24 compute P_Psychological=Psychological*Perpetrator1/co.
25 compute P_Self_control=Self_control*Perpetrator1/co.
26 compute P_Attachment=Attachment*Perpetrator1/co.
27 compute P_Passion=Passion*Perpetrator1/co.
28 compute P_Perpetrator=Perpetrator*Perpetrator1/co.
29 compute P_Delinquency=Delinquency*Perpetrator1/co.
30 compute P_pred=Perpetrator1.
31 execute.
32 compute V_Strain=Strain*Victim/co.
33 compute V_Physical=Physical*Victim/co.
34 compute V_Psychological=Psychological*Victim/co.
35 compute V_Self_control=Self_control*Victim/co.
36 compute V_Attachment=Attachment*Victim/co.
37 compute V_Passion=Passion*Victim/co.
38 compute V_Perpetrator=Perpetrator*Victim/co.
39 compute V_Delinquency=Delinquency*Victim/co.
40 compute V_pred=Victim.
41 execute.
***
```

Analysis: We investigated the effect of the cyber bullying input factors on the output factors (prediction probability). In the cyber bullying neural-network prediction model, the various types had the following prediction probabilities, with the following factor influences.

- Perpetrator prediction probability: 12.33%. Factor influence: perpetrator (offender_psychology) (4.54%), strain (3.11%), delinquency (1.65%), attachment (1.21%), passion (0.71%), physical (0.66%), psychological (victim_psychology) (0.35%), and self-control (0.1%).
- Victim prediction probability: 70.57%. Factor influence: strain (19.3%), attachment (14.48%), delinquency (11.68%), physical (10.48%), passion (8.14%), psychological (5.24%), self-control (1.24%), and perpetrator (0.0%).
- Bystander prediction probability: 9.4%. Factor influence: strain (2.66%), attachment (2.03%), passion (1.65%), delinquency (1.52%), physical (0.75%), psychological (0.60%), self-control (0.20%), and perpetrator (0.0%).
- Complex prediction probability: 7.7%. Factor influence: strain (1.91%), attachment (1.91%), passion (1.23%), delinquency (1.15%), physical (0.71%), psychological (0.64%), self-control (0.16%), and perpetrator (0.0%).

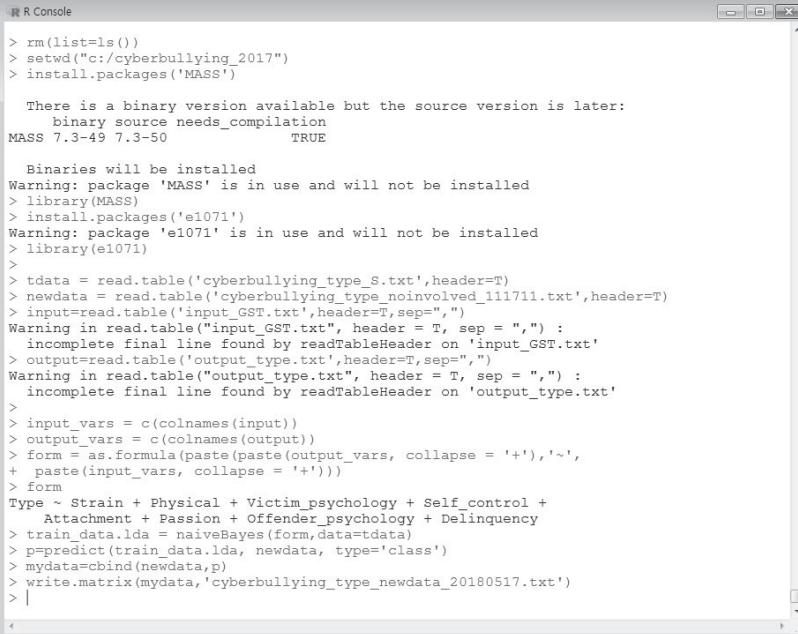
	N	Mean									
P_Strain	59073	.0311	V_Strain	59073	.1930	B_Strain	59073	.0266	C_Strain	59073	.0191
P_Physical	59073	.0066	V_Physical	59073	.1048	B_Physical	59073	.0075	C_Physical	59073	.0071
P_Psychological	59073	.0035	V_Psychological	59073	.0524	B_Psychological	59073	.0060	C_Psychological	59073	.0064
P_Self_control	59073	.0010	V_Self_control	59073	.0124	B_Self_control	59073	.0020	C_Self_control	59073	.0016
P_Attachment	59073	.0121	V_Attachment	59073	.1448	B_Attachment	59073	.0203	C_Attachment	59073	.0191
P_Passion	59073	.0071	V_Passion	59073	.0814	B_Passion	59073	.0165	C_Passion	59073	.0123
P_Perpetrator	59073	.0454	V_Perpetrator	59073	.0000	B_Perpetrator	59073	.0000	C_Perpetrator	59073	.0000
P_Delinquency	59073	.0165	V_Delinquency	59073	.1168	B_Delinquency	59073	.0152	C_Delinquency	59073	.0115
P_pred	59073	.1233	V_pred	59073	.7057	B_pred	59073	.0940	C_pred	59073	.0770
Valid N (listwise)	59073		Valid N (listwise)	59073		Valid N (listwise)	59073		Valid N (listwise)	59073	

Using Training Data with Input Variables to Create Dependent Variables

We used training data with input variables but no dependent variables to create dependent variables as predicted by the Naïve Bayes classification model, and added them to the training data. The training data for machine learning training in this study (data with both independent and

dependent variables: 59,672) can be referenced to perform training and modeling. Then, the Naïve Bayes classification model that was developed can be used to predict dependent variables for the 111,711 documents that have only independent variables (GST factors) and no independent variables (cyber bullying types: perpetrator, victim, bystander, complex) among the cyber bullying data collected in this study.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("e1071")
> library(e1071)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_type_S.txt',header=T)
  - Assign the learning data file containing dependent and independent
    variables to the tdata object.
> newdata = read.table('cyberbullying_type_noinvolved_111711.txt',
  header=T)
  - Assign the data containing only the independent variable to the
    newdata object.
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_type.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+'))))
> form
> train_data.lda = naiveBayes(form,data=tdata)
  - Create a model function (train_data.lda) by executing the Naive
    Bayes classification model with tdata set.
> p=predict(train_data.lda, newdata, type='class')
  - Create a predictive group of cyber school violence types by executing
    model predictions with a newdata dataset that has no dependent
    variables.
> mydata=cbind(newdata,p)
  - Assign to the mydata object, including the predicted dependent
    variable (p) and the variable for newdata.
> write.matrix(mydata,'cyberbullying_type_newdata_20180517.txt')
  - Save the mydata object as 'cyberbullying_type_newdata_
    20180517 .txt' file.
```



The screenshot shows the R console window with the following text:

```

> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('MASS')
There is a binary version available but the source version is later:
  binary source needs_compilation
MASS 7.3-49 7.3-50 TRUE

Binaries will be installed
Warning: package 'MASS' is in use and will not be installed
> library(MASS)
> install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
> library(e1071)
>
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> newdata = read.table('cyberbullying_type_noinvolved_111711.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning in read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
> train_data.lda = naiveBayes(form,data=tdata)
> p=predict(train_data.lda, newdata, type='class')
> mydata=cbind(newdata,p)
> write.matrix(mydata,'cyberbullying_type_newdata_20180517.txt')
> |

```

```

> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('catspec')
> library(catspec)
> new_data = read.table('cyberbullying_type_newdata_20180517.txt',
  header=T)
  - Look at the included type of cyber-school violence (p) predicted by
    the Naive Bayes classification model.
> attach(new_data)
> t1=ftable(new_data[c('p')]): Type frequency is calculated.
> ctab(t1,type=c('n','r'))
> length(p)

```

Type	Strain	Physical	Victim_psychology	Self_control	Attachment	Passion	Offender_psychology	Delinquency	p
5	1	1	1	0	1	1	0	0	Victim
5	0	1	0	0	1	1	0	0	Victim
5	1	0	1	0	1	0	1	0	Victim
5	1	0	0	0	0	1	0	0	Victim
5	0	1	0	0	1	0	0	0	Victim
5	1	0	0	0	1	1	0	0	Victim
5	1	0	0	0	1	0	0	1	Victim
5	1	0	1	0	1	1	0	0	Victim
5	1	0	0	0	0	1	0	0	Victim
5	1	0	0	0	1	0	0	0	Victim
5	1	0	0	0	1	1	0	0	Victim
5	1	0	0	0	1	0	0	0	Victim
5	1	0	0	0	1	0	0	0	Victim
5	1	0	0	0	1	1	0	1	Victim
5	1	0	0	0	1	1	0	0	Victim
5	1	0	0	0	1	1	0	0	Victim
5	0	0	0	0	0	1	0	0	Victim
5	0	1	0	0	1	1	0	0	Victim
5	1	0	0	0	1	1	1	1	Victim
5	1	0	0	1	1	1	1	0	Complex
5	1	0	0	0	0	1	0	0	Victim
5	0	1	0	0	1	1	0	0	Victim
5	0	1	0	0	0	0	0	0	Victim
5	1	0	0	0	0	0	0	1	Perpetrator
5	0	1	0	0	1	0	0	1	Victim

```
R Console
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('catspec')
Warning: package 'catspec' is in use and will not be installed
> library(catspec)
> new_data = read.table('cyberbullying_type_newdata_20180517.txt',header=T)
> #attach(new_data)
> t1=ftable(new_data[c('p')])
> ctab(t1,type=c('n','r'))
  x Complex Perpetrator   Victim
Count      3595.00    27208.00 80908.00
Total %     3.22       24.36   72.43
> length(p)
[1] 111711
> |
```

The cyber bullying types (p) predicted by the Naïve Bayes classification model, using data that contained only independent variable factors, included the following: Perpetrator: 24.36% (27,208), victim: 72.43% (80,908), complex: 3.22% (3,595). Therefore, if these data are added to existing training data (59,672), it is possible to create 171,383 new training data (59,672+111,711).

Creating Data with the Same Training-Data and Predicted-Data Classifications

We created data with the same training data classification and predicted data classification. High-quality training data were created by selecting cases where the cyber bullying risk (negative, positive) classification included in this study's training data and the predicted classification (negative, positive), which was made using the developed neural network, were the same. It can be seen that 109,908 high-quality training data were created.

```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("nnet")
> library(nnet)
> install.packages('MASS')
> library(MASS)
> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
- 152,426 cases (negative: 45.32%, positive: 54.68%) are saved in the
  tdata object.
> input=read.table('input_GST.txt',header=T,sep=",")
> output=read.table('output_attitude.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
> paste(input_vars, collapse = '+'))))
> form
> tr.nnet = nnet(form, data=tdata, size=5)
> p=predict(tr.nnet, tdata, type='class')
> mydata=cbind(tdata, p)
> write.matrix(mydata,'cyberbullying_attitude_predict_20180517.txt')
```



The screenshot shows the R Console window with the following R script execution:

```

R R Console
> library(nnet)
> install.packages('MASS')

There is a binary version available but the source version is later:
  binary source needs_compilation
MASS 7.3-49 7.3-50          TRUE

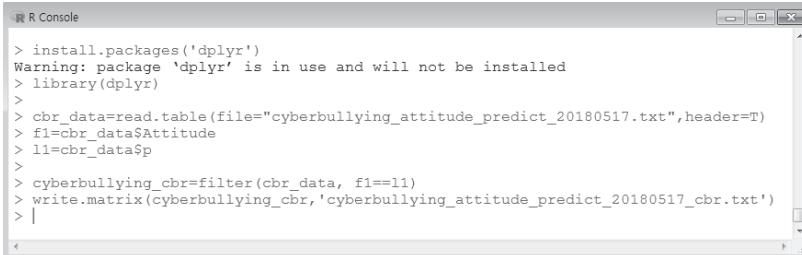
Binaries will be installed
Warning: package 'MASS' is in use and will not be installed
> library(MASS)
>
> tdata = read.table("cyberbullying_attitude_S.txt", header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning in read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning in read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> tr.nnet = nnet(form, data=tdata, size=5)
# weights:  51
initial value 117781.005985
iter  10 value 88266.995795
iter  20 value 86624.784793
iter  30 value 86392.922451
iter  40 value 86201.774165
iter  50 value 86087.156946
iter  60 value 86009.301930
iter  70 value 85965.003965
iter  80 value 85948.908257
iter  90 value 85940.461611
iter 100 value 85931.149395
final value 85931.149395
stopped after 100 iterations
> p=predict(tr.nnet, tdata, type='class')
> mydata=cbind(tdata, p)
> write.matrix(mydata,'cyberbullying_attitude_predict_20180517.txt')
> |

```

```

> install.packages('dplyr')
> library(dplyr)
> cbr_data=read.table(file= "cyberbullying_attitude_predict_20180517.txt", header=T)
> f1=cbr_data$Attitude
> l1=cbr_data$p
> cyberbullying_cbr=filter(cbr_data, f1==l1)
> write.matrix(cyberbullying_cbr,'cyberbullying_attitude_predict_20180517_cbr.txt')

```



```

R Console

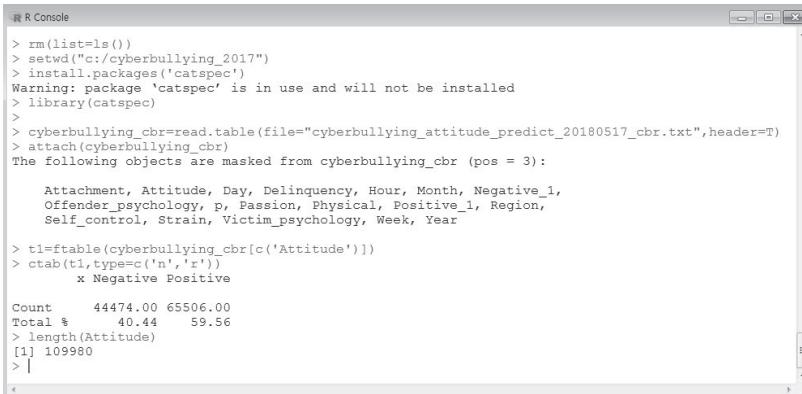
> install.packages('dplyr')
Warning: package 'dplyr' is in use and will not be installed
> library(dplyr)
>
> cbr_data=read.table(file="cyberbullying_attitude_predict_20180517.txt",header=T)
> f1=cbr_data$Attitude
> l1=cbr_data$p
>
> cyberbullying_cbr=filter(cbr_data, f1==l1)
> write.matrix(cyberbullying_cbr,'cyberbullying_attitude_predict_20180517_cbr.txt')
> |

```

```

> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('catspec')
> library(catspec)
> cyberbullying_cbr=read.table(file="cyberbullying_attitude_predict_
  20180517_cbr.txt", header=T)
> attach(cyberbullying_cbr)
> t1=ftable(cyberbullying_cbr[c('Attitude')])
> ctab(t1,type=c('n','r'))
> length(Attitude)

```



```

R Console

> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('catspec')
Warning: package 'catspec' is in use and will not be installed
> library(catspec)
>
> cyberbullying_cbr=read.table(file="cyberbullying_attitude_predict_20180517_cbr.txt",header=T)
> attach(cyberbullying_cbr)
The following objects are masked from cyberbullying_cbr (pos = 3):
  Attachment, Attitude, Day, Delinquency, Hour, Month, Negative_1,
  Offender_psychology, p, Passion, Physical, Positive_1, Region,
  Self_control, Strain, Victim_psychology, Week, Year

> t1=ftable(cyberbullying_cbr[c('Attitude')])
> ctab(t1,type=c('n','r'))
  x Negative Positive

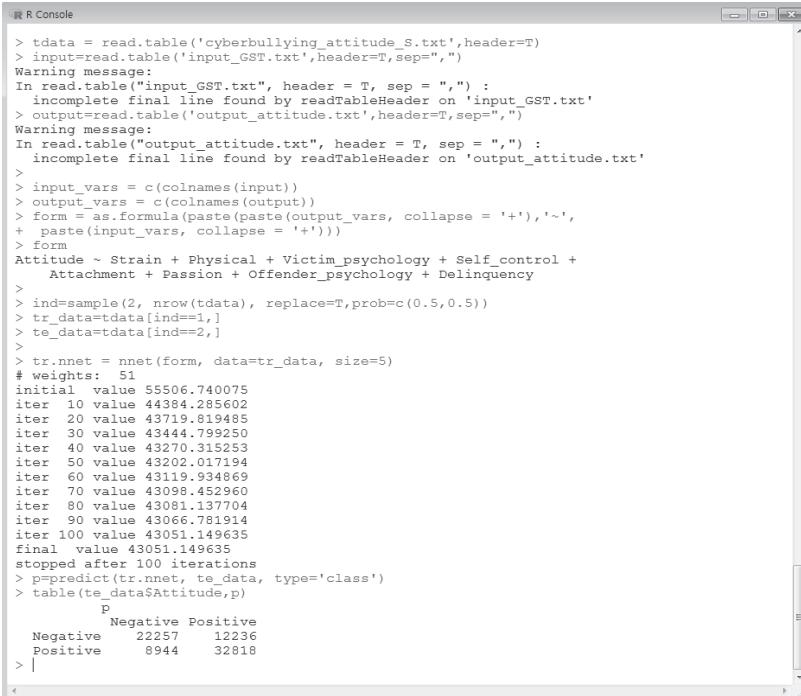
Count      44474.00 65506.00
Total %    40.44   59.56
> length(Attitude)
[1] 109980
> |

```

Evaluating Existing Training Data and High Quality Training Data

High-quality training data can be created by selecting data for which this study's training-data classification and the neural network model's predicted classification are the same. A model evaluation of the existing training data and the high-quality training data is as follows.

■ Existing Training Data's Neural Network Model Evaluation

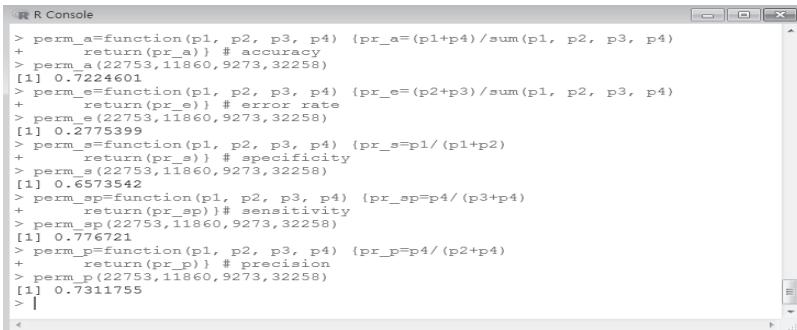


```

R Console

> tdata = read.table('cyberbullying_attitude_S.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=", ")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=", ")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
>
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> tr.nnet = nnet(form, data=tr_data, size=5)
# weights:  51
initial value 55506.740075
iter 10 value 44384.285602
iter 20 value 43719.819485
iter 30 value 43444.799250
iter 40 value 43270.315253
iter 50 value 43202.017194
iter 60 value 43119.934869
iter 70 value 43098.452960
iter 80 value 43081.137704
iter 90 value 43066.781914
iter 100 value 43051.149635
final value 43051.149635
stopped after 100 iterations
> p=predict(tr.nnet, te_data, type='class')
> table(te_data$Attitude,p)
   p
      Negative Positive
Negative    22257    12236
Positive     8944    32818
> |

```



```

> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+ return(pr_a)} # accuracy
> perm_a(22753,11860,9273,32258)
[1] 0.7224601
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+ return(pr_e)} # error rate
> perm_e(22753,11860,9273,32258)
[1] 0.2775399
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+ return(pr_s)} # specificity
> perm_s(22753,11860,9273,32258)
[1] 0.6573542
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+ return(pr_sp)}# sensitivity
> perm_sp(22753,11860,9273,32258)
[1] 0.776721
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+ return(pr_p)} # precision
> perm_p(22753,11860,9273,32258)
[1] 0.7311755
> |

```

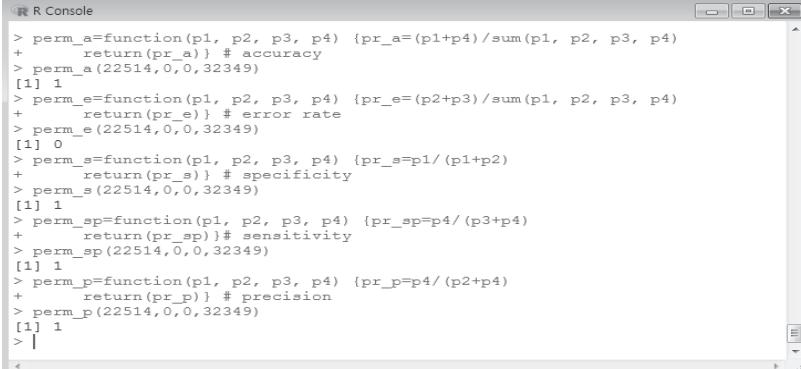
■ High Quality Training Data's Neural Network Model Evaluation



```

> tdata = read.table('cyberbullying_attitude_predict_20180517_cbr.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
>
> tr.nnet = nnet(form, data=tr_data, size=5)
# weights:  51
initial value 44935.635227
iter 10 value 5288.071102
iter 20 value 2891.243447
iter 30 value 438.446494
iter 40 value 73.181470
iter 50 value 0.134102
iter 60 value 0.000251
final value 0.000087
converged
> p=predict(tr.nnet, te_data, type='class')
> table(te_data$Attitude,p)
   P
Negative 22166      0
Positive      0 32761
> |

```



```

> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(22514,0,0,32349)
[1] 1
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(22514,0,0,32349)
[1] 0
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(22514,0,0,32349)
[1] 1
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)}# sensitivity
> perm_sp(22514,0,0,32349)
[1] 1
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(22514,0,0,32349)
[1] 1
> |

```

The existing training data's model-evaluation results showed an accuracy of 72.22%. The high quality training data's model evaluation results showed an accuracy of 100.0%. Therefore, it is believed that this training data has excellent model evaluation results. The prediction accuracy can be increased by predicting new data after developing a machine learning model (artificial intelligence).

Creating an Artificial Intelligence with Machine Learning

■ Predicting Cyber bullying Types

A Naïve Bayes classification model function for predicting cyber bullying types using this study's training data can be used to predict the cyber bullying types for data that contains only input variables.

```

> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('MASS')
> library(MASS)
> install.packages('e1071')
> library(e1071)
> tdata = read.table("cyberbullying_type_S.txt",header=T)

```

Year	Month	Day	Hour	Week	Region	Type	Perpetrator_1	Victim_1	Bystander_1	Complex_1	Strain	Physical	Victim_psychology	Self_control	Attachment	Passion	Offender_psychology	Delinquency	.00
2013.00	11.00	30.00	8.00	토	.00	Perpetrator	1.00	.00	.00	.00	1.00	.00	.00	1.00	.00	1.00	.00	.00	
2013.00	11.00	22.00	7.00	금	.00	Victim	.00	1.00	.00	.00	.00	.00	.00	1.00	.00	.00	.00	.00	
2013.00	11.00	6.00	14.00	수	.00	Bystander	.00	.00	1.00	.00	1.00	.00	.00	.00	1.00	1.00	.00	1.00	
2013.00	11.00	4.00	8.00	화	.00	Complex	.00	.00	.00	1.00	1.00	.00	.00	.00	.00	.00	.00	.00	
2013.00	9.00	13.00	13.00	금	.00	Victim	.00	1.00	.00	.00	1.00	.00	.00	.00	1.00	.00	.00	.00	
2013.00	9.00	2.00	8.00	화	.00	Complex	.00	.00	.00	1.00	.00	.00	.00	1.00	1.00	.00	1.00	.00	
2013.00	8.00	13.00	7.00	화	대구	Bystander	.00	.00	1.00	.00	.00	1.00	.00	.00	1.00	.00	.00	.00	
2013.00	7.00	8.00	21.00	금	.00	Victim	.00	1.00	.00	.00	1.00	.00	.00	.00	1.00	1.00	.00	.00	
2013.00	6.00	2.00	21.00	금	.00	Complex	.00	.00	.00	1.00	1.00	.00	.00	.00	.00	1.00	.00	.00	
2013.00	6.00	14.00	7.00	금	경기	Victim	.00	1.00	.00	.00	1.00	.00	.00	.00	1.00	1.00	1.00	.00	
2013.00	5.00	24.00	10.00	목	전남	Victim	.00	1.00	.00	.00	1.00	.00	.00	1.00	1.00	1.00	1.00	.00	
2013.00	4.00	25.00	9.00	화	.00	Perpetrator	1.00	.00	.00	.00	1.00	.00	.00	1.00	1.00	.00	.00	.00	

```
> newdata = read.table('cyberbullying_type_newdata_AI_20180517.txt',
  header=T)
```

- Assign six data that include only input variables to newdata.

```
R Console
> newdata
  Type Strain Physical Victim_psychology Self_control Attachment Passion Offender_psychology Delinquency
1  NO      1       1             1          0           1     1        0            0
2  NO      0       1             0          0           1     1        0            0
3  NO      1       0             0          1           1     1        1            0
4  NO      1       0             0          0           0     0        0            1
5  NO      0       0             0          1           1     0        0            0
6  NO      1       0             0          0           0     0        0            1
> |
```

```
> input=read.table('input_GST_n.txt',header=T,sep=",")
> output=read.table('output_type.txt',header=T,sep=",")
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
  paste(input_vars, collapse = '+')))
> form
> train_data.lda = naiveBayes(form,data=tdata)
  - Create a machine learning model function (artificial intelligence:
    train_data.lda).
> p=predict(train_data.lda, newdata, type='class')
  - Use the model function (train_data.lda) that predicts cyberbullying
    types to predict school cyberbullying types in newdata and assign
    them to object p.
> mydata=cbind(newdata,p)
  - Add the predicted cyber bullying types to the newdata dataset and
    save it in mydata.
> write.matrix(mydata,'cyberbullying_type_predict_AI_20180517_1.txt')
  - It can be seen that the predicted types (p) were added to the data in
    newdata, which does not have school cyberbullying types (NO).
```



The screenshot shows two R console windows. The top window displays a data frame named 'mydata' with columns: Type, Strain, Physical, Victim_psychology, Self_control, Attachment, Passion, Offender_psychology, Delinquency, and P. The bottom window shows the R code used to load data and train a Naïve Bayes model.

```

R Console
> mydata
   Type Strain Physical Victim_psychology Self_control Attachment Passion Offender_psychology Delinquency      P
1  NO     1         1                 1          0          1       1          0          0      0      0    Victim
2  NO     0         1                 0          0          1       1          1          0      0      0    Victim
3  NO     1         0                 0          0          1       1       1          1      1      0    Complex
4  NO     1         0                 0          0          0       0          0          0      0      1 Perpetrator
5  NO     0         0                 0          1          1       1          0          0      0      0    Complex
6  NO     1         0                 0          0          0       0          0          0      0      1 Perpetrator
> |

R Console
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages('MASS')

There is a binary version available but the source version is later:
  binary source needs_compilation
MASS 7.3-49 7.3-50           TRUE

Binaries will be installed
Warning: package 'MASS' is in use and will not be installed
> library(MASS)
> install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
> library(e1071)
> tdata = read.table('cyberbullying_type_S.txt',header=T)
> newdata = read.table('cyberbullying_type_newdata_AI_20180517.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning message:
In read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
> #newdata
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
> train_data.lda = naiveBayes(form,data=tdata)
> p=predict(train_data.lda, newdata, type='class')
> mydata=cbind(newdata,p)
> write.matrix(mydata,'cyberbullying_type_predict_AI_20180517_1.txt')
> mydata

```

■ Calculating Prediction-Probability Values

A Naïve Bayes classification model function for predicting cyber bullying types using this study's training data can be used to calculate the prediction probability values of each cyber bullying type for data that contain only input variables.



```

> install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
> library(e1071)
> tdata = read.table('cyberbullying_type_N_AI.txt',header=T)
> newdata = read.table("cyberbullying_type_predict_AI_20180517_1.txt",header=T)
> input_GST=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_type.txt',header=T,sep=",")
Warning message:
In read.table("output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_type.txt'
> p_output=read.table('p_output_type.txt',header=T,sep=",")
Warning message:
In read.table("p_output_type.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_type.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Type ~ Strain + Physical + Victim_psychology + Self_control +
Attachment + Passion + Offender_psychology + Delinquency
> train_data=cbind(input,newdata,form,data=tdata)
> predict(train_data,id=newdata, type="raw")
> dimnames(p)=list(NULL,c(p.output_vars))
> pred_obs = cbind(newdata, p)
> write.matrix(pred_obs,'cyberbullying_type_naiveBayes_AI_prob.txt')
> mydata1=read.table('cyberbullying_type_naiveBayes_AI_prob.txt',header=T)
> mydata1
   Type Strain Physical Victim_psychology Self_control Attachment Passion Offender_psychology Delinquency
1  NO      1       1           1      0      1      1      0      0
2  NO      0       1           0      0      1      1      1      0
3  NO      1       0           0      1      1      1      1      0
4  NO      1       0           0      0      0      0      0      1
5  NO      0       0           0      1      1      0      0      0
6  NO      1       0           0      0      0      0      0      1
> |          p_Perpetrator p_Victim p_Bystander p_Complex
1  Victim  8.967548e-03  0.8076899  0.07071771  0.11262487
2  Victim  3.165481e-02  0.8321758  0.07088556  0.06528378
3  Complex 6.7825799e-06  0.1375022  0.20331046  0.65918058
4 Perpetrator 5.762525e-01  0.3576661  0.04558304  0.02049837
5  Complex 3.534009e-04  0.3919625  0.16471733  0.44296678
6 Perpetrator 5.762525e-01  0.3576661  0.04558304  0.02049837
>

```

It can be seen that the predicted probabilities (p) for each predicted type were added to the data in newdata, which do not have cyber bullying types (NO). Therefore, The first document is 80.76% predicted by Victim, the second document is 83.22% predicted by Victim, the third document is 65.92% predicted to be complex, the fourth document is 57.63% predicted by perpetrator, The probability that a document will be predicted as a complex is 44.29%, and the probability that a sixth document will be predicted as a perpetrator is 57.63%.

- Predict cyber bullying attitude(negative/positive) using high quality training data

Using the neural network model function, the cyber bullying attitude can be predicted for the data containing only input variables.

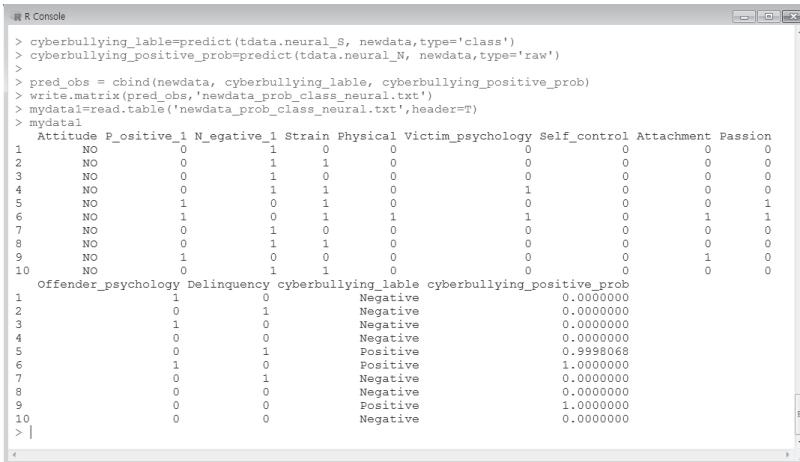
R Console

```
> # Utilizing Artificial Intelligence (Estimation of Probability Values)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> install.packages("nnet")
Warning: package 'nnet' is in use and will not be installed
> library(nnet)
> install.packages('MASS')
Warning: package 'MASS' is in use and will not be installed
> library(MASS)
>
> tSdata = read.table('cyberbullying_attitude_predict_20180517_cbr_S.txt',header=T)
> tNdata = read.table('cyberbullying_attitude_predict_20180517_cbr_N.txt',header=T)
> input=read.table('input_GST.txt',header=T,sep=",")
Warning message:
In read.table("input_GST.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'input_GST.txt'
> output=read.table('output_attitude.txt',header=T,sep=",")
Warning message:
In read.table("output_attitude.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_attitude.txt'
> newdata = read.table('cyberbullying_attitude_newdata_20180629.txt',header=T)
>
> # neural networks modeling
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Strain + Physical + Victim_psychology + Self_control +
  Attachment + Passion + Offender_psychology + Delinquency
> |
```

R Console

```
> tdata.neural_S = nnet(form, data=tSdata, size=5)
# weights:  51
initial value 76158.581297
iter 10 value 3085.710800
iter 20 value 1690.869836
iter 30 value 570.936889
iter 40 value 156.878803
iter 50 value 8.082655
iter 60 value 0.644122
iter 70 value 0.140880
iter 80 value 0.019336
iter 90 value 0.003697
iter 100 value 0.001600
final value 0.001600
stopped after 100 iterations
> tdata.neural_N = nnet(form, data=tNdata, size=5)
# weights:  51
initial value 31899.186037
iter 10 value 1996.332384
iter 20 value 525.993411
iter 30 value 121.880372
iter 40 value 15.791181
iter 50 value 0.235185
iter 60 value 0.009604
final value 0.000065
converged
> p=predict(tdata.neural_S, tSdata,type='class')
> table(tSdata$Attitude,p)

          P
Negative    44474      0
Positive       0    65506
> |
```

A screenshot of an R console window titled "R Console". The window shows R code being run and its output. The code involves predicting cyberbullying labels and positive probabilities for a dataset named "newdata". The output displays two tables: one for predicted labels and one for predicted probabilities. The first table has columns for various attitude variables and their corresponding predicted labels (1 for Positive, 0 for Negative). The second table lists the predicted probability of being Positive for each document, ranging from 0.000000 to 0.9998068.

```
> cyberbullying_lable=predict(tdata.neural_S, newdata,type='class')
> cyberbullying_positive_prob=predict(tdata.neural_N, newdata,type='raw')
>
> pred_obs = cbind(newdata, cyberbullying_lable, cyberbullying_positive_prob)
> write.matrix(pred_obs,'newdata_prob_class_neural.txt')
> mydata1=read.table('newdata_prob_class_neural.txt',header=T)
> mydata1
  Attitude P_positive_1 N_negative_1 Strain Physical_Victim_psychology Self_control Attachment Passion
1      NO          0           1          0          0          0          0          0          0
2      NO          0           1           1          0          0          0          0          0
3      NO          0           1           0          0          0          0          0          0
4      NO          0           1           1          0          0          1          0          0
5      NO          1           0           1          0          0          0          0          0
6      NO          1           0           0           1          1          0          0          1
7      NO          0           1           0          0          0          0          0          0
8      NO          0           1           1          0          0          0          0          0
9      NO          1           0           0          0          0          0          0          1
10     NO          0           1           1          0          0          0          0          0
Offender_psychology Delinquency cyberbullying_lable cyberbullying_positive_prob
1                  1          0             Negative 0.0000000
2                  0          1             Negative 0.0000000
3                  1          0             Negative 0.0000000
4                  0          0             Negative 0.0000000
5                  0          1             Positive 0.9998068
6                  1          0             Positive 1.0000000
7                  0          1             Negative 0.0000000
8                  0          0             Negative 0.0000000
9                  0          0             Positive 1.0000000
10                 0          0             Negative 0.0000000
> |
```

It can be seen that the predicted attitude's classification (cyberbullying_lable) and the attitude's prediction probability (cyberbullying_positive_prob) are added to the newdata data with no negative or positive cyber bullying (NO). Therefore, the first document is classified as Negative and the probability of being predicted as Positive is 0%. The fifth document was classified as positive and the probability of being predicted as positive was 99.98%.

VISUALIZATION

Data visualization refers to the process of visually expressing and communicating data analysis results or machine learning prediction results so that they can be understood easily. Visualization includes text data visualization, time-series data visualization, and geographical data visualization.

Visualization of Text Data

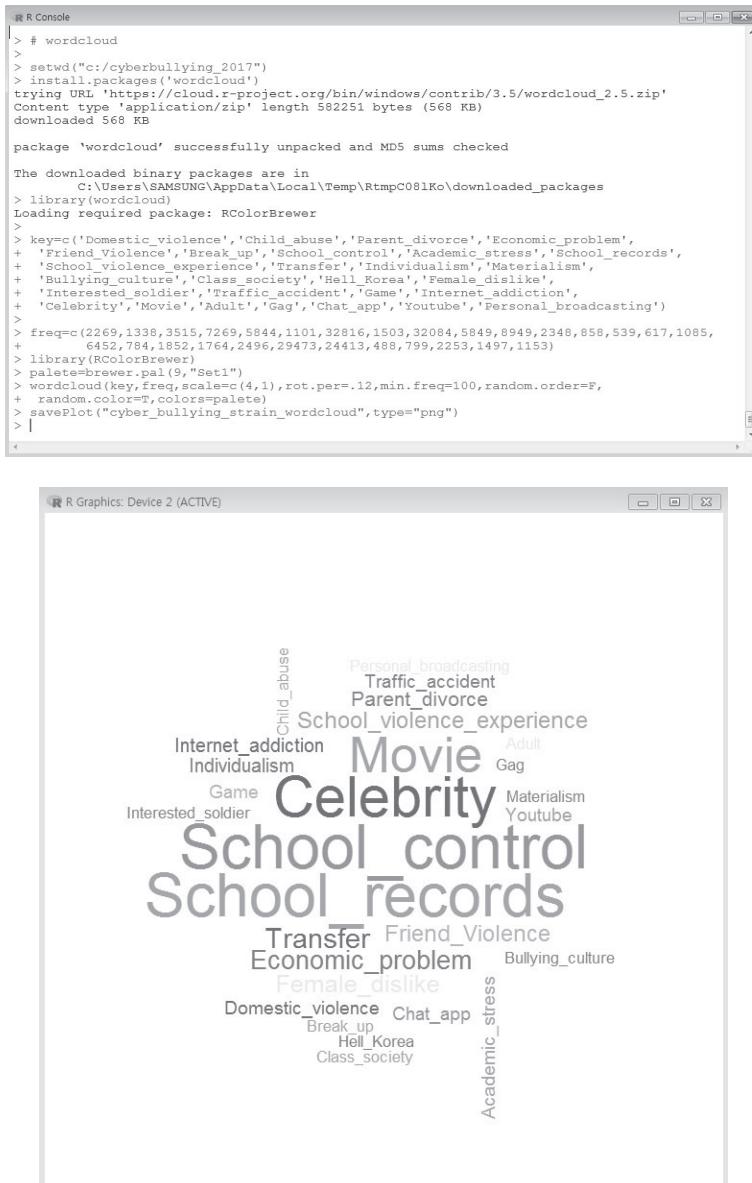
Word clouds are often used to visualize text data. A word cloud is a visual technique that shows words from a big-data text database as a cloud shape in a 2D space, making it easy to see the frequency at which the words appear. Generally, the character's size is proportional to the frequency, and if a word is extremely frequent, it is positioned toward the center.

- Creating a word cloud for cyber bullying strain factors.

```
> setwd("c:/cyberbullying_2017"): Set the working directory.  
> install.packages('wordcloud')  
  - Install the Word Cloud package.  
> library(wordcloud): Load the Word Cloud package.  
> key=c('Domestic_violence','Child_abuse','Parent_divorce','Economic_problem','Friend_Violence','Break_up','School_control','Academic_stress','School_records','School_violence_experience','Transfer','Individualism','Materialism','Bullying_culture','Class_society','Hell_Korea','Female_dislike','Interested_soldier','Traffic_accident','Game','Internet_addiction','Celebrity','Movie','Adult','Gag','Chat_app','Youtube','Personal_broadcasting')  
  - Assign the cyber bullying strain-factor keywords to the key vector.  
> 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 the cyber bullying strain factor keyword frequencies to the frequency vector.  
> library(RColorBrewer): Load the package that displays the color.
```

```
> palete=brewer.pal(9,"Set1")
> wordcloud(key,freq,scale=c(4,1),rot.per=.12,min.freq=100,random.order
  =F,random.color=T,colors=palete): Produce the word cloud.
  - key: Shows the words (text) assigned to the key vector.
  - freq: Shows the frequencies of the words assigned to the freq vector.
  - scale(4, 1): Shows the word size (maximum 4, minimum 1). The
    default value is c(4, 0.5).
  - rot.per = .12: Display and arrange the 12% of the words assigned to
    the key vector at 90 degrees.
  - min.freq = 100: Set the minimum number of mentions (display only
    words mentioned 100 times or more). The default value is 3.
  - max.words: Set the number of words to be displayed. The default is
    to display all words. If max.words is set, words are displayed in
    descending order until the number is reached.
  - random.order = F: Position words from the screen's center to the
    edge, according to the drawing order. If the argument is T, the words
    are drawn in random order. If the argument is F, the words are
    arranged in descending order of frequency. If F is chosen, the words
    with the highest appearance frequencies are placed in the center. The
    default value is T.
  - random.color = T: If the argument is F, the word's color is set
    according to the color order set in the colors argument in descending
    order of frequency. If the argument is T, the color is set randomly.
    The default value is F.
  - colors = palette: Set the colors of words that are displayed by
    frequency. Sets the colors of displayed words to the colors assigned to
    the palette variable.
```

> savePlot("cyber_bullying_strain_wordcloud",type="png")
- Save the results in an image file.



Analysis: The cyber bullying strain-factor keywords were concentrated on School control, Celebrity, School records, and Movie.

Visualization of Time Series Data

Big data in time series format can be visualized as a line graph or a bar graph.

Line Graph Visualization

Line graphs use the `plot()` function. The `plot()` function's main arguments are shown in Table 6.

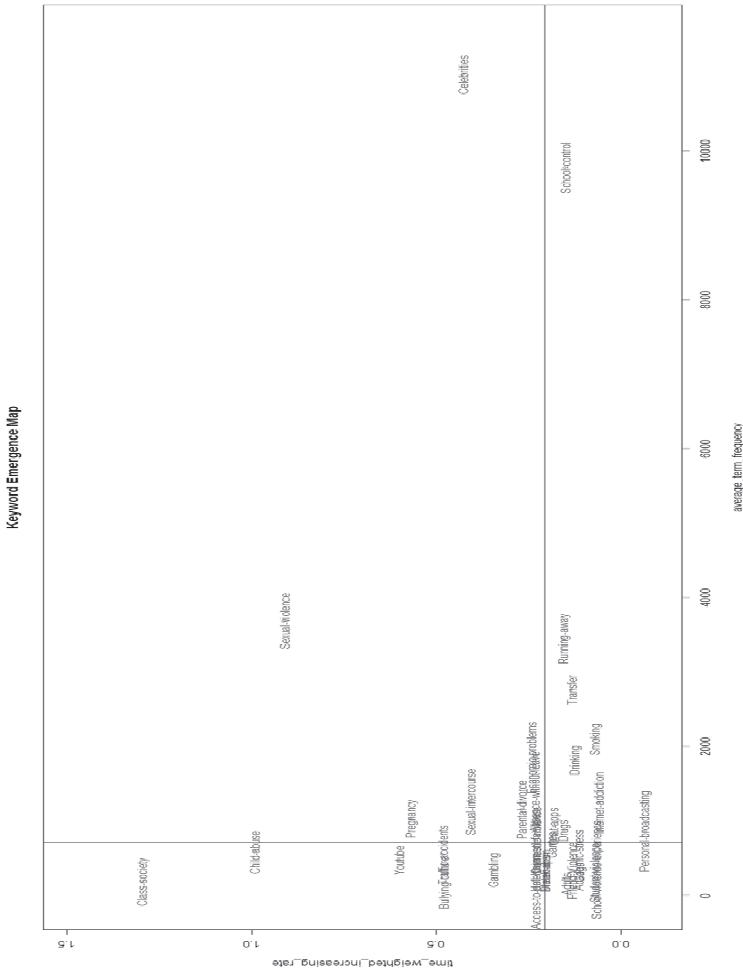
Table 6 `plot()` function's main arguments

arguments	function
<code>type='p'</code>	plot should be drawn[points(p), lines(l), points/lines(b), the lines part alone of "b"(c), both 'overplotted'(o), stair steps(s)]
<code>xlim</code>	the x limits of the plot
<code>ylim</code>	the y limits of the plot
<code>log='x'</code>	a character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic
<code>main</code>	a main title for the plot
<code>sub</code>	a sub title for the plot
<code>xlab</code>	a label for the x axis, defaults to a description of x
<code>ylab</code>	a label for the y axis, defaults to a description of y
<code>ann</code>	a logical value indicating whether the default annotation (title and x and y axis labels) should appear on the plot
<code>axes</code>	a logical value indicating whether both axes should be drawn on the plot
<code>col='blue'</code> , <code>col=4</code>	The colors for lines and points[black(1), red(2), green(3), blue(4), cyan(5), magenta(6) yellow(7), gray(8), etc.]
<code>lty</code>	a vector of line types[(0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash)]
<code>las</code>	the style of axis labels[always parallel to the axis (0), always horizontal(1), always perpendicular to the axis(2), always vertical(3)]
<code>lwd</code>	a vector of line widths(a positive number, defaulting to 1)
<code>cex</code>	a numerical vector giving the amount by which plotting characters and symbols should be scaled relative to the default
<code>font</code>	An integer which specifies which font to use for text(1=plain, 2:bold, 3=italic, 4=bold italic, 5=symbol)
<code>pch</code>	a vector of plotting characters or symbols[□(0), ○(1), Δ(2), +(3), X(4), etc]

■ Cyber bullying Delinquency and Strain Factor Future Signals Search Visualization (degree of visibility (DoV))

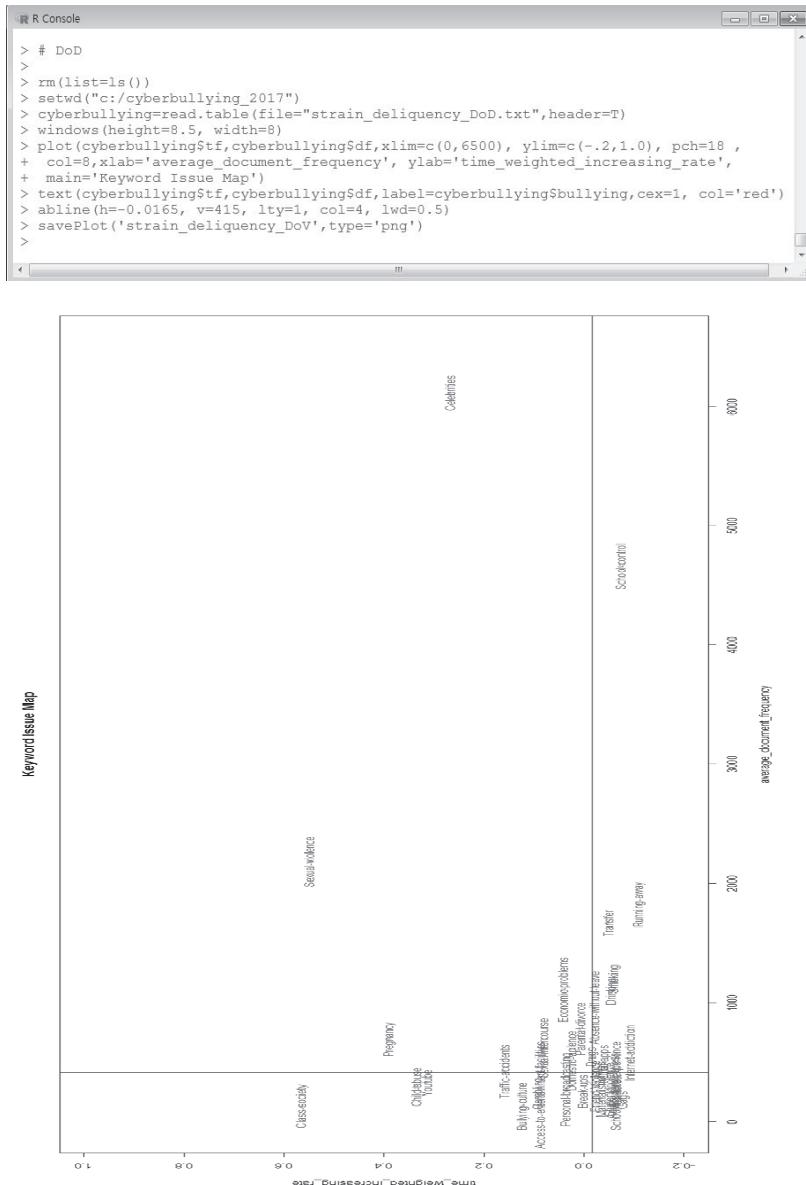
```
> rm(list=ls()): Initialize all variables.
> setwd("c:/cyberbullying_2017"): Set the working directory.
> cyberbullying=read.table(file="strain_deliqency_DoV.txt",header=T)
  - Assign data to the cyberbullying variable.
> windows(height=8.5, width=8)
  - A graphics device is opened (Thus, fixate the display size).
> plot(cyberbullying$tf,cyberbullying$df,xlim=c(0,11500),ylim=c(-.1,1.5),
  pch=18,col=8,xlab='average_term_frequency',ylab='time_weighted_
  increasing_rate', main='Keyword Emergence Map')
  - cyberbullying$tf: Set the variable cyberbullying$tf on the x-axis.
  - cyberbullying$df: Set the variable cyberbullying$df on the y axis.
  - xlim=c(0,11500): the x limits(0,11500) of the plot.
  - ylim=c(-.1,1.5): the y limits(-.1,1.5) of the plot.
  - pch=18: a vector of plotting characters or symbols(0=square ~
    18=filled diamond blue2).
  - col=8: The colors for points[black(1), red(2), green(3), blue(4),
    cyan(5), magenta(6),yellow(7), gray(8)].
  - xlab=' ': a label for the x axis.
  - ylab=' ': a label for the y axis.
  - main=' ': a main title for the plot.
> text(cyberbullying$tf,cyberbullying$df,label=cyberbullying$bullying,
  cex=0.8, col='red')
  - text(): text draws the strings given in the vector labels at the
    coordinates given by x and y.
  - Display the 'cyberbullying$bullying' in red with the 0.8 text size,
    which are located on the x(cyberbullying$tf) and
    y(tf,cyberbullying$df) of the coordinator chart.
> abline(h=0.206, v=712, lty=1, col=4, lwd=0.5)
  - abline(): This function adds one or more straight lines through the
    current plot.
  - h=horizon, v=vertical, lty=a vector of line types(solid ), col=The
    colors for lines and points(blue), lwd: a vector of line widths.
> savePlot('strain_deliqency_DoV',type='png')
  - Save the result as a picture file (type = jpg, png, pdf, etc.).
```

² <http://www.endmemo.com/program/R/pchsymbols.php>



Analysis: In the KEM (Keyword Emergence Map), the cyber bullying strain and delinquency factors' strong signals were sexual violence, celebrities, pregnancy, sexual intercourse, parental divorce, absence without leave, economic problems, and domestic violence. The weak signals were class society, child abuse, YouTube, bullying culture, traffic accidents, gambling, access to entertainment facilities, materialism, Hell Korea, and break-ups. The latent signals were games, academic stress, school violence experience, friend violence, gags, student violence, and adults. The signals that were strong but had a low rate of increase were chat apps, drugs, internet addiction, personal broadcasting, running away, transfer, smoking, drinking, and school control.

- Cyber bullying Delinquency and Strain Factors' Future-Signal Search Visualization (degree of diffusion (DoD))

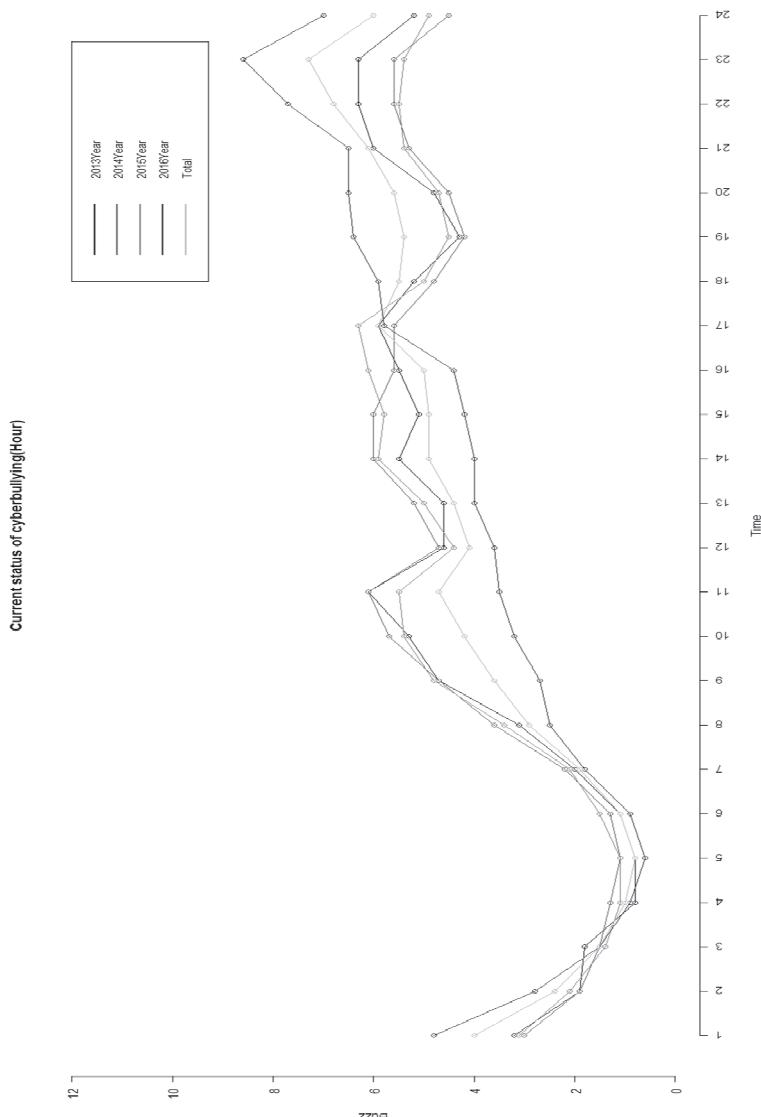


Analysis: In the KIM (Keyword Issue Map), the cyber bullying strain and delinquency factors' strong signals were sexual violence, celebrities, pregnancy, traffic accidents, sexual intercourse, parental divorce, economic problems, domestic violence, and drugs. The weak signals were class society, child abuse, YouTube, bullying culture, gambling, access to entertainment facilities, Hell Korea, break-ups, and personal broadcasting. The latent signals were games, academic stress, school violence experience, friend violence, gags, student violence, adults, and materialism. The signals that were strong but had a low rate of increase were chat apps, internet addiction, running away, transfer, smoking, drinking, school control, and absence without leave.

■ Visualization of Cyber bullying Document Status by Time of Day

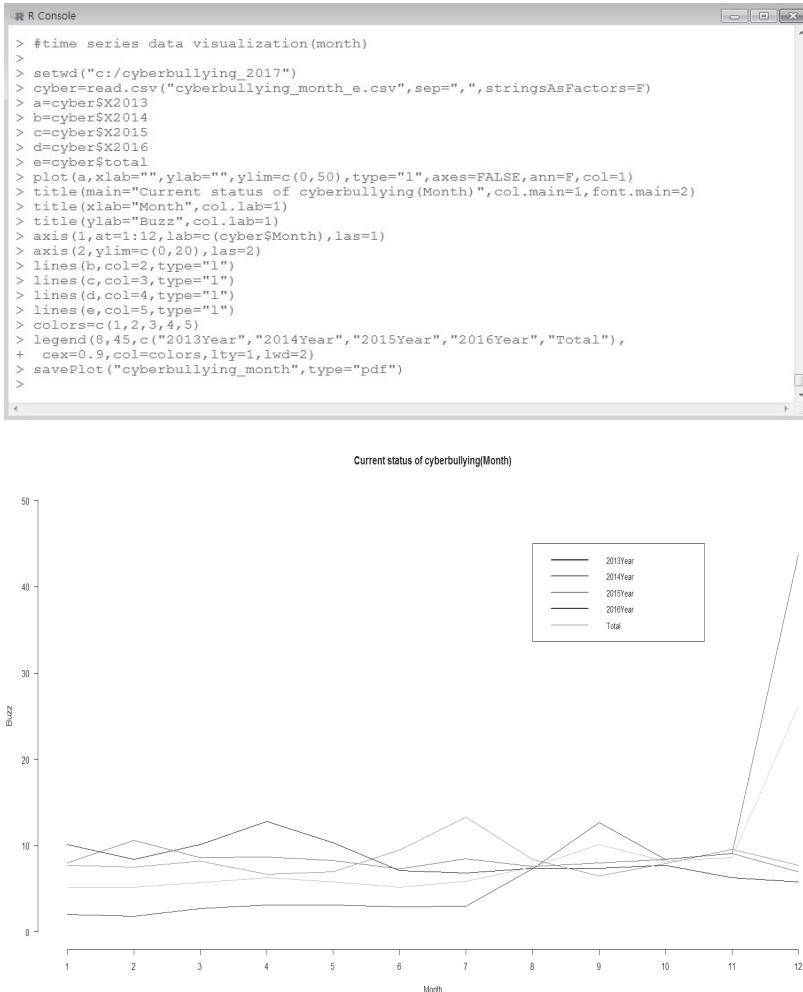
```
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> cyber=read.csv("cyberbullying_time_e.csv",sep=",",stringsAsFactors=F)
> a=cyber$X2013year: Assign the 'A' variable to the 2013 content.
> b=cyber$X2014year
> c=cyber$X2015year
> d=cyber$X2016year
> e=cyber$total: Assign the total item to the 'A' variable.
> plot(a,xlab="",ylab="",ylim=c(0,12),type="o",axes=FALSE,ann=F,col=1)
    - a: Set the variable "a" to which the item is assigned in 2013.
    - xlab="Time", ylab="Buzz": a label for the x axis, defaults to a
      description of x and y.
    - ylim=c(0, 12): the y limits(0~12) of the ploty.
    - type="o": plot should be drawn(both overplotted).
    - axes=FALSE: Do not mark both x and y axes.
    - ann=F: Do not assign a title to both x and y axes.
    - col=1: The colors for lines[black(1), red(2), green(3), blue(4),
      cyan(5), magenta(6) yellow(7), gray(8), etc].
> title(main="Current status of cyberbullying(Hour)",col.main=1,font.main=2)
    - main="Main title": a main title for the plot
    - col.main=1: The colors for title[1: black].
    - font.main=2: font to use for title[1=plain, 2=bold, 3=italic, 4=bold
      italic, 5=symbol ].
> title(xlab="Time",col.lab=1): Set the x-axis string to black.
> title(ylab="Buzz",col.lab=1): Set the y-axis string to black.
    - Buzz: 'word of mouth' means online document.
```

```
> axis(1,at=1:24,lab=c(cyber$Time),las=2)
  - Displays the x and y axes as a specified value.
  - 1: Set the axes (1: x-axis, 2: y-axis).
  - at=1:24: Set the range (1 to 24) of the x-axis.
  - lab=c(cyber$ Time): Display the time item of cyber data on the screen.
  - las=2 Create labels (items) on the x-axis perpendicular to the axes (1: horizon, 2: vertical).
> axis(2,ylim=c(0,12),las=2): Display the x and y axes as a specified value.
  - 2: Set the axes (1: x-axis, 2: y-axis).
  - ylim = c (0, 12): Set the range (1 to 12) on the y-axis.
  - las = 2: Create a label (item) on the y-axis perpendicular to the axes.
> lines(b,col=2,type="o")
  - Display the red dotted lines as overlap on the screen on 2014.
> lines(c,col=3,type="o")
  - Display the green dotted lines as overlap on the screen in 2015.
> lines(d,col=4,type="o")
  - Display the blue dotted lines as overlap on the screen on 2016.
> lines(e,col=5,type="o")
  - Display the total on the screen in a dotted light blue lines as overlap.
> colors=c(1,2,3,4,5): Designate the color of the contents.
> legend(18,12,c("2014Year","2015Year","2016Year","Total"),
  cex=0.9,col=colors,lty=1,lwd=2): Designate the formant of the contents.
  - legend(18, 12): Designate the location of the contents (18th x-axis and 12th y-axis).
  - c("2013year" ~ "Total"): Display the contents of the legend.
  - cex=0.9: Set the text size of the legend.
  - col=colors: Set the color of the legend [c(1, 2, 3, 4, 5)].
  - lty=1: Designate the line type (1: line).
  - lwd=2: Designate the line width.
> savePlot("cyberbullying_time.pdf",type="pdf")
  - Save the result as a picture.
```



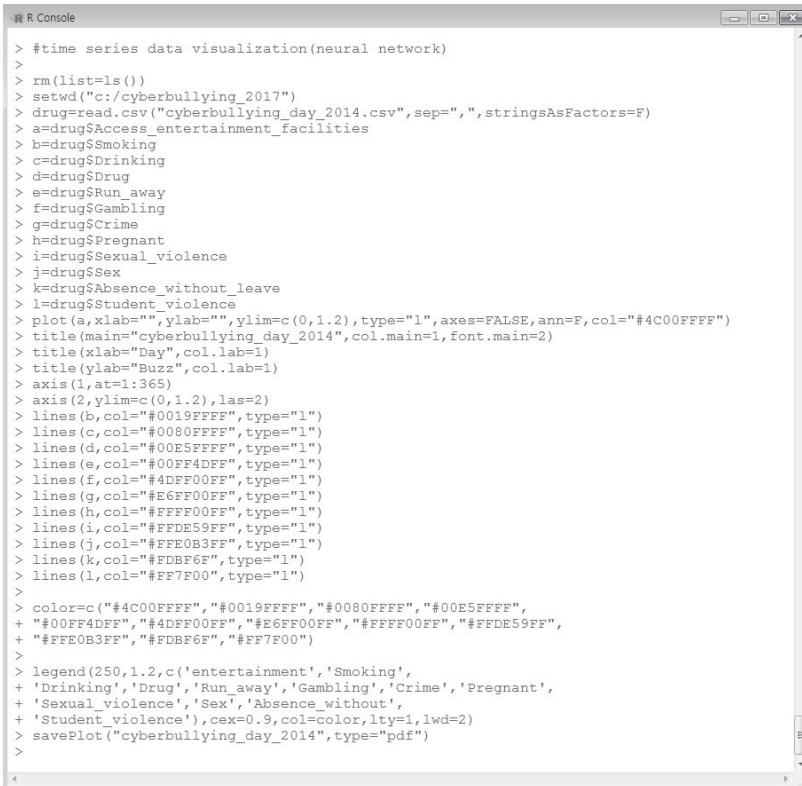
Analysis: Aside from 2016, online documents related to cyber bullying showed an increasing trend starting at 7 AM when students go to school. They decreased after 11 AM, increased after 12 PM, decreased again after 5 PM, increased after 7 PM, and decreased after 11 PM.

■ Visualization of Cyber bullying Document Status by Month



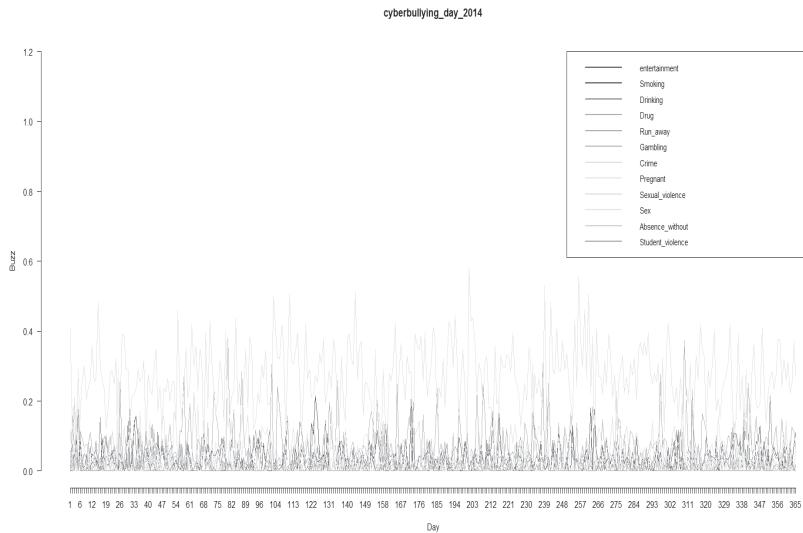
Analysis: With the exception of 2016, online documents related to cyber bullying showed an increasing trend beginning in March when school starts and decreasing in April. They increased in June, decreased in July, increased in September, and decreased in November.

■ Visualization of Cyber bullying Delinquency Factor Risk by Day (Neural Network)



The screenshot shows the R Console window with the following R script:

```
> #time series data visualization(neural network)
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> drug=read.csv("cyberbullying_day_2014.csv",sep=",",stringsAsFactors=F)
> a=drug$Access_entertainment_facilities
> b=drug$Smoking
> c=drug$Drinking
> d=drug$Drug
> e=drug$Run_away
> f=drug$Gambling
> g=drug$Crime
> h=drug$Pregnant
> i=drug$Sexual_violence
> j=drug$Sex
> k=drug$Absence_without_leave
> l=drug$Student_violence
> plot(a,xlab="",ylab="",ylim=c(0,1.2),type="l",axes=FALSE,ann=F,col="#4C00FFFF")
> title(main="cyberbullying_day_2014",col.main=1,font.main=2)
> title(xlab="Day",col.lab=1)
> title(ylab="Buzz",col.lab=1)
> axis(1,at=1:365)
> axis(2,ylim=c(0,1.2),las=2)
> lines(b,col="#0019FFFF",type="l")
> lines(c,col="#0080FFFF",type="l")
> lines(d,col="#00E5FFFF",type="l")
> lines(e,col="#00FF4DF",type="l")
> lines(f,col="#4DFE00FF",type="l")
> lines(g,col="#E6FF00FF",type="l")
> lines(h,col="#FFF000FF",type="l")
> lines(i,col="#FDE59FF",type="l")
> lines(j,col="#FFE0B3FF",type="l")
> lines(k,col="#FDBF6F",type="l")
> lines(l,col="#FFTFO0",type="l")
>
> color=c("#4C00FFFF", "#0019FFFF", "#0080FFFF", "#00E5FFFF",
+ "#00FF4DF", "#4DFE00FF", "#E6FF00FF", "#FFF000FF",
+ "#FDE59FF", "#FFE0B3FF", "#FDBF6F", "#FFTFO0")
>
> legend(250,1.2,c('entertainment','Smoking',
+ 'Drinking','Drug','Run_away','Gambling','Crime','Pregnant',
+ 'Sexual violence','Sex','Absence_without',
+ 'Student violence'),cex=0.9,col=color,lty=1,lwd=2)
> savePlot("cyberbullying_day_2014",type="pdf")
>
```



Analysis: The daily average of the neural network model's cyber bullying risk prediction probabilities are as follows. Drinking daily average of 3.39%, Drug daily average of 1.37%, Run_away daily average of 6.74%, Gambling daily average of 0.67%, Crime daily average of 26.84%, Pregnant , Daily average of sexual activity was 0.97%, Sexual_violence was 4.35%, Sex was daily average 1.42%, Absence_without_leave was 2.69% daily, Student violence was 1.42% and daily risk was 53.91%.

Bar Graph Visualization

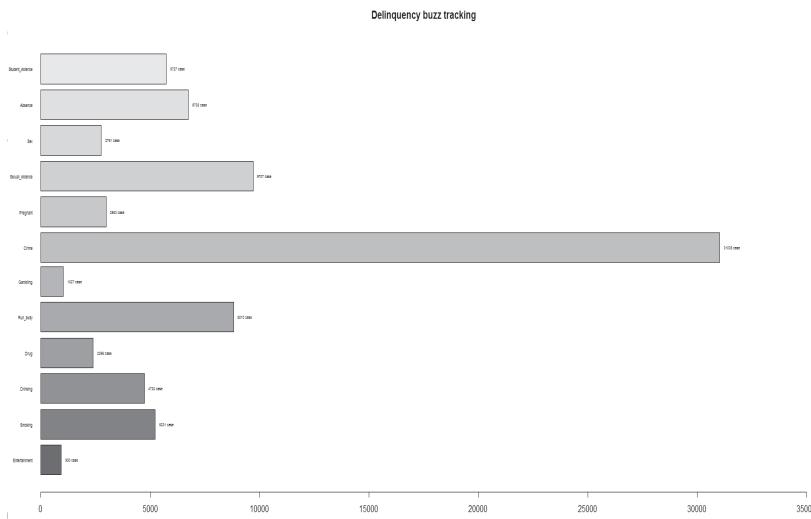
■ Bar Graph Visualization (Delinquency Factors' Appearance Frequency)

The bar graph uses the barplot() function.

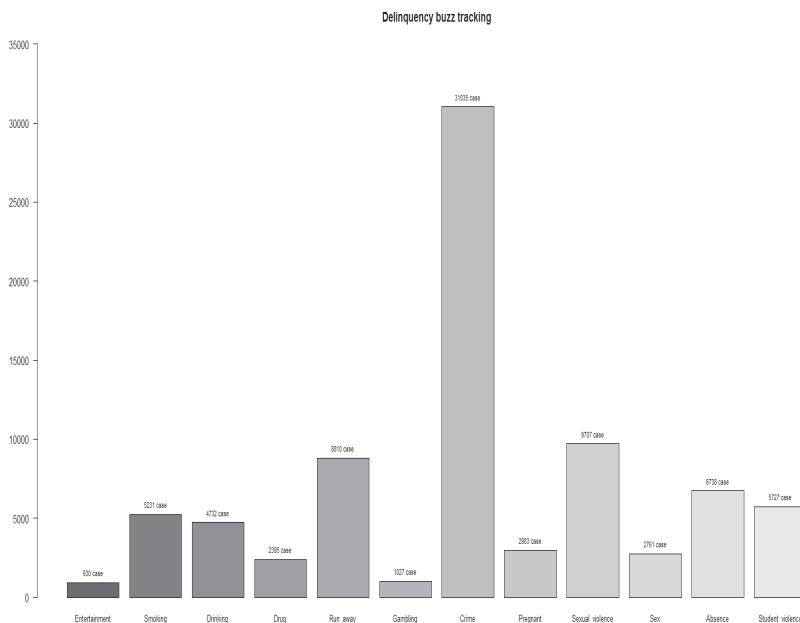
```
> rm(list=ls()): Initialize all variables.
> setwd("c:/cyberbullying_2017"): Set the working directory.
> f=read.csv("cyberbullying_delinquency.csv",sep=",",stringsAsFactors=F)
> bp=barplot(f$Frequency, names.arg=f$delinquency,main="Delinquency
  buzz tracking",col=gray.colors(12),xlim=c(0,35000),cex.names=0.5,
  col.main=1,font.main=2,las=1,horiz=T)
  - f$Frequency: Set the x-axis variable (frequency).
  - names.arg=f$delinquency: Set the x-axis variable (delinquency).
  - main="Delinquency buzz tracking": Set the title of the graph.
```

- col=gray.colors(12): Specify 12 colors in black and white.
- xlim=c(0,35000): Set the range of the x-axis.
- cex.names=0.7: Set the font size of the y-axis.
- col.main=1: Set the graph title to black.
- font.main=2: Set the font of the graph title (1: dark, 2: light, 3: tilted).
- las=1: draw the label of the axis horizontally (las = 2: vertical).
- horiz=T : Set the bar graph horizontally (F or default is vertical).

```
> text(y=bp,x=f$Frequency*1,pos=4,labels=paste(f$Frequency,'case'),
  col='black',cex=0.4)
  - pos=1(below), 2(left), 3(above, default), 4(right).
  - Display the variable frequency bar (black, 0.5 size) of the y-axis on
  the horizon.
```



```
R Console
> # barPlot visualization
>
> rm(list=ls())
> setwd("c:/cyberbullying_2017")
> f=read.csv("cyberbullying_delinquency.csv",sep=",",stringsAsFactors=F)
> bp=barplot(f$Frequency, names.arg=f$delinquency,main="Delinquency buzz tracking",
+ col=gray.colors(12),ylim=c(0,35000),cex.names=0.8,
+ col.main=1,font.main=2,las=1,horiz=T)
> # pos=1(below), 2(left), 3(above, default), 4(right)
> text(x=bp,y=f$Frequency*1,pos=3,labels=paste(f$Frequency,'case'),
+ col='black',cex=0.7)
>
```



Visualization of Geographical Data

Geographical data or spatial data gathered from big data can be analyzed through visualization to observe the associations between regions, and changes in regions over time. Visualizations of United States geographical information in R can use the “maps” and “mapproj” packages.

■ Drawing Geographical Maps at a State Level

The map data included in the “maps” package includes information at the level of states and counties, which are subunits of states. The USArrests data frame contains information on the latitude and longitude coordinates of American states, as well as information on assailants in the 48 states of continental United States, excluding Alaska and Hawaii, in the year 1973. Therefore, to show recent statistics on the map (e.g., the number of murderers by state in 2015), the states’ coordinates and the recent statistics to be shown on the map must be merged in the USArrests data frame.

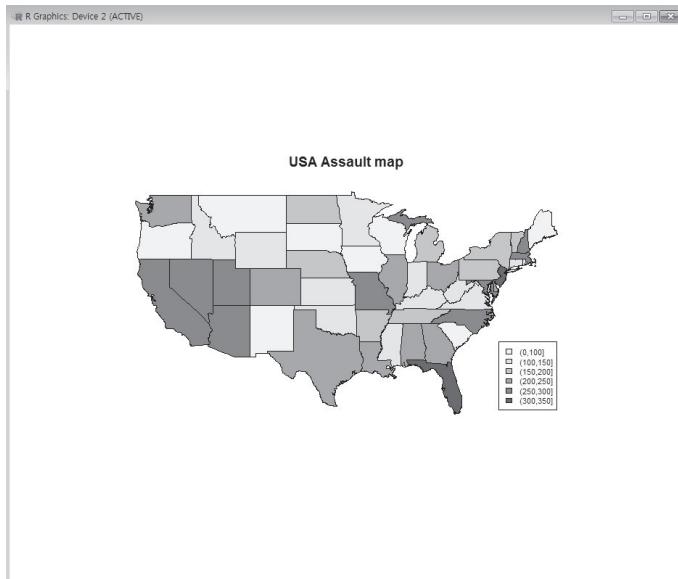
Research Problem: Show the 2015 statistics for the number of murders on the map of the 48 states of continental United States.

```
> setwd("c:/cyberbullying_2017")
> install.packages('maps')
  - Install the package (maps) for visualizing geographical data.
> library(maps):Load the maps package.
  - data reference: https://ucr.fbi.gov/crime-in-the-u.s/2015/crime-in-the-u.s.-2015/tables/table-4.
  - Save the number of murders by state from the site that shows crime for 2015 in the United States.
> USAArrests_Murder=c(348,309,181,1861,176,117,63,1041,615,32,744,373,72,
  128,209,481,23,516,128,571,133,259,502,36,62,178,14,363,117,609,517,21,500,
  234,99,658,29,399,32,406,1316,54,10,383,211,70,240,16)
  - Assign the number of murders in the 48 states in 2015 to USAArrests_Murder as a vector. The above vector value can be modified for crime statistics for the 48 states for other years.
> usa_sub=subset(USAArrests,!rownames(USAArrests)%in%c('Alaska','Hawaii'))
  - Assign the USAArrests data frame information (excluding Alaska and Hawaii, which are separate from the mainland) to usa_sub.
> usa_data_Murder=data.frame(states=rownames(usa_sub),Murder=USAArrests_Murder)
  - Merge the state map coordinates and the 2015 murder numbers in usa_sub and assign it to usa_data_Murder.
> usa_data_Murder
  - Display the usa_data_Murder data frame on the screen.
> color.level=cut(usa_data_Murder[,2],c(0, 50, 100, 200, 300, 500, 1000,
  2000))
  - Create the legend data for each state, considering usa_data_Murder's second variable (Murder) values, and assign it to color.level. The legend of each state was divided into seven intervals [(0,50) - (1000, 2000)] to create the legend data for each state.
> Legend=levels(color.level)
  - Assign the legend with seven intervals to be included in the map to Legend.
> levels(color.level)=rev(heat.colors(7))
  - Assign each interval's color according to the heat gradient.
  - heat.colors, topo.colors, cm.colors, terrain.colors, rainbow, diverge.colors, gray.colors.
  - The heat function creates a gradient palette that goes from dark colors to light colors; thus, to show darker colors on the map as the statistical values increase, the rev function must be used to change the order.
```

```
> color.level=as.character(color.level)
- Assign the hex values of the colors corresponding to the legend of
each state to color.level.
> map('state', region = usa_data$states, fill=TRUE, col=color.level)
- Show the statistics on the number of murders by state on the map.
> title("USA murders by state, 2015"): Set the map's title.
> legend(-77,34,Legend, fill=rev(heat.colors(7)), cex=0.7)
- Set the legend with a font size of 0.7 at coordinates (-77, 34) in an
empty space where the state map is not being displayed.
```

The screenshot shows the R Console window with the following code:

```
> setwd("c:/cyberbullying_2017")
> install.packages('maps')
Warning: package 'maps' is in use and will not be installed
> library(maps)
> install.packages('maps')
Warning: package 'maps' is in use and will not be installed
> library(maps)
> # data reference: https://ucr.fbi.gov/crime-in-the-u-s/2015/crime-in-the-u-s--2015/tables/table-4
> # Murder
>
> USArrests_arrest=c(236,263,294,190,276,204,110,238,335,211,46,120,249,113,56,115,109,249,
+ 83,300,72,259,178,109,102,252,57,159,285,254,337,45,120,151,159,106,174,279,86,188,
+ 201,120,48,156,145,81,53,161)
> #USArrests
> usa_sub=subset(USArrests,!rownames(USArrests)%in% c('alaska','Hawaii'))
> usa_data=data.frame(states=rownames(usa_sub),Assault=USArrests_arrest)
> #usa_data
> col.level=cut(usa_sub[,2],c(0,100,150,200,250,300,350))
> Legend=levels(col.level)
> levels(col.level)=rev(heat.colors(6))
> col.level=as.character(col.level)
> map('state', region = usa_data$states, fill=TRUE, col=col.level)
> title("USA Assault map")
> legend(-75,33,Legend, fill=rev(heat.colors(6)), cex=0.7)
> |
```



■ Drawing Geographical Maps at a County Level

- reference:
<http://bcb.dfci.harvard.edu/~aedin/courses/R/CDC/maps.html>

Research Problem: Show United States county-level statistics (e.g., 2009 unemployment) on a map.

```
> install.packages('MASS')
  - Install the MASS package, which includes the write.matrix() function.
> library(MASS): Load the MASS package.
> install.packages('mapproj')
  - Install the mapproj package, which supports the projection function.
> library(mapproj): Load the mapproj package.
> data(unemp)
  - Import the county-level map coordinate information (Federal
    Information Processing Standard (FIPS) county code) and the
    unemp data, including the 2009 unemployment information
    [population (pop), unemployment (unemp)], into the R data frame.
> data(county.fips)
  - Import the county.fips data, which includes the county-level map-
    coordinate information (FIPS code) and the corresponding county
```

names (polynname), into the R data frame.

```
> write.matrix(unemp,'unemp.txt')
  - Save the unemp data as unemp.txt so it can be modified with recent statistics.
```

```
> unemp=read.table('unemp.txt',header=T)
  - Assign the unemp.txt file to unemp.
```

```
> colors=c('#F1EEF6','#D4B9DA','#C994C7','#DF65B0','#DD1C77','#980043')
  - Assign the colors that will show the unemployment rate legend for each county to colors as hex values.
```

```
> unemp$colorBuckets=as.numeric(cut(unemp$unemp,c(0,2,4,6,8,10,100)))
  - Divide the legend for each county into six intervals [(0, 2) - (10, 100)] and assign them to unemp$colorBuckets.
```

```
> leg.txt=c('<2%','2>4%','4>6%','6>8%','8>10%','>10%')
  - Assign the six-interval legend that will be included in the map to leg.txt.
```

```
> cnty.fips=county.fips$fips[match(map('county',plot=FALSE)$names,
  county.fips$polynname)]
  - Assign the FIPS data, which shows the map coordinates of each county, to cnty.fips.
```

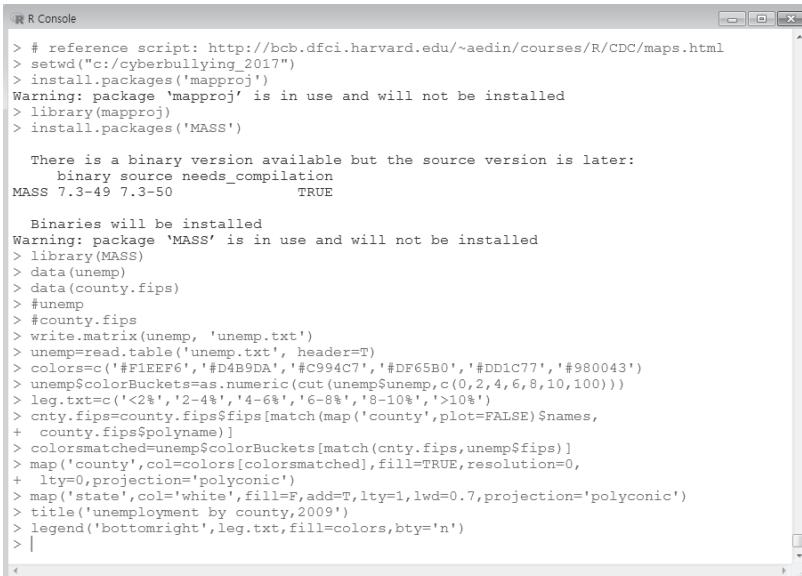
```
> colorsmatched=unemp$colorBuckets[match(cnty.fips,unemp$fips)]
  - Assign the color numbers that correspond to the unemployment rate for each county to colorsmatched.
```

```
> map('county',col=colors[colorsmatched],fill=TRUE,resolution=0,
  lty=0,projection='polyconic')
  - Draw the map so that it is filled in with the colors that correspond to the legend for the unemployment rate of each county.
```

```
> map('state',col='white',fill=F,add=T,lty=1,lwd=0.7,projection='polyconic')
  - Project the states' boundary lines in white.
```

```
> title('unemployment by county, 2009'): Set the title of the map.
```

```
> legend('bottomright',leg.txt,fill=colors,bty='n', cex=0.9): Set the legend.
```

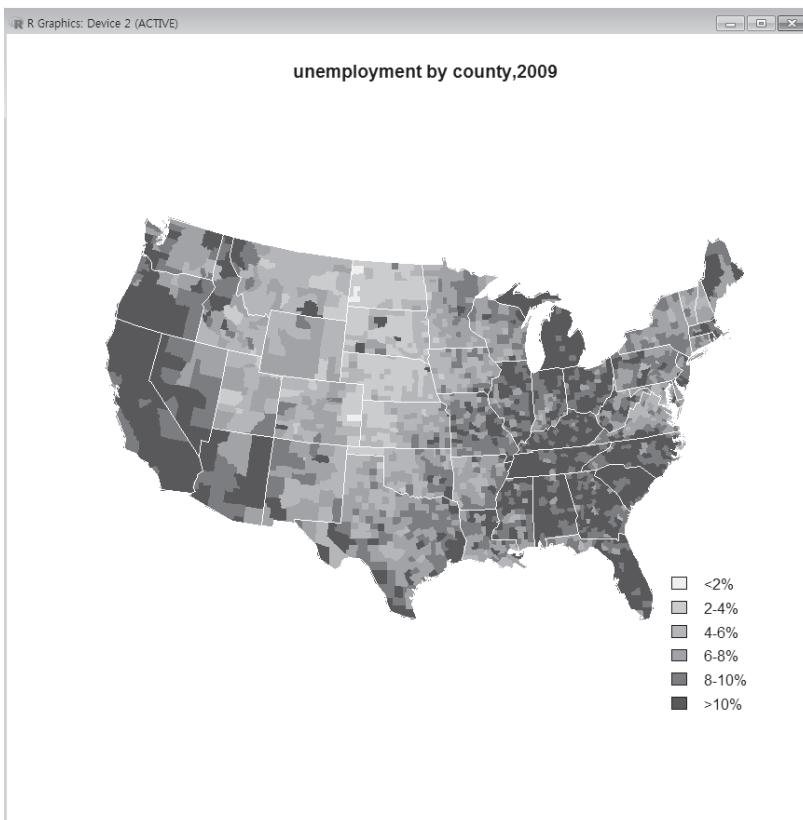


The screenshot shows an R console window titled "R Console". The window contains R script code for analyzing unemployment data by county. The code includes several library imports, data loading from a file named "unemp.txt", and the creation of a choropleth map using the "map" function from the "maptools" package. The map displays county-level unemployment rates in 2009, with colors ranging from light blue (low rates) to dark red (high rates). A legend is provided in the bottom right corner of the plot area.

```
> # reference script: http://bcb.dfc.harvard.edu/~aedin/courses/R/CDC/maps.html
> setwd("c:/cyberbullying_2017")
> install.packages('mapproj')
Warning: package 'mapproj' is in use and will not be installed
> library(mapproj)
> install.packages('MASS')

There is a binary version available but the source version is later:
  binary source needs_compilation
MASS 7.3-49 7.3-50          TRUE

Binaries will be installed
Warning: package 'MASS' is in use and will not be installed
> library(MASS)
> data(unemp)
> data(county.fips)
> #unemp
> #county.fips
> write.matrix(unemp, 'unemp.txt')
> unemp=read.table('unemp.txt', header=T)
> colors=c('#F1EEF6','#D4B9DA','#C994C7','#DF65B0','#DD1C77','#980043')
> unemp$colorBuckets<-as.numeric(cut(unemp$unemp,c(0,2,4,6,8,10,100)))
> leg.txt=c('<2%','2-4%','4-6%','6-8%','8-10%','>10%')
> cnty.fips=county.fips$fips[match(map('county',plot=FALSE)$names,
+ county.fips$polyname)]
> colorsmatched=unemp$colorBuckets[match(cnty.fips,unemp$unemp)]
> map('county',col=colors[colorsmatched],fill=TRUE,resolution=0,
+ lty=0,projection='polyconic')
> map('state',col='white',fill=F,add=T,lty=1,lwd=0.7,projection='polyconic')
> title('unemployment by county, 2009')
> legend('bottomright',leg.txt,fill=colors,bty='n')
> |
```



References

- Berk, R. A., & Bleich, J. (2014). Forecasts of violence to inform sentencing decisions. *Journal of Quantitative Criminology*, 30, 79-96.
- Breiman, L. (1996). Bagging predictors, *Machine Learning*, 26, 123-140.
- . (2001). Random forest, *Machine learning*, 45(1), 5-32.
- Cortes, C & Vapnik, V (1995). Support-vector networks, *machine Learning*, 20, 273-297.
- David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams. Learning representations by back-propagating errors. 1986, *Nature*, 323:533-536.
- Duwe, G., & Kim, K. (2017). Out with the old and in with the new? An empirical comparison of supervised learning algorithms to predict recidivism. *Criminal Justice Policy Review*, 28(6), 570-600.

- Greiner M., Pfeiffer, D., Smith RD. (2000). Principles and practical application of the receiver-operating characteristic analysis for diagnostic tests. *J Preventive Veterinary Medicine*, 45(1-2), 23-41.
- Hand, D., Mannila, H., Smyth P., "Principles of Data Mining.", The MIT Press, 2001, Cambridge, ML.
- Jin, J. H., Oh, M. A. (2013). Data Analysis of Hospitalization of Patients with Automobile Insurance and Health Insurance: A Report on the Patient Survey. *Journal of the Korea Data Analysis Society*, 15(5B), 2457-2469.
- Minsky M, Papert S, "Perceptrons.", 1969, MIT Press, Cambridge.
- Mitchell, Tom. M. 1997. Machine Learning. New York: McGraw-Hill., 59.
- Park, H. C. (2013). Proposition of causal association rule thresholds. *Journal of the Korean Data & Information Science Society*, 24(6), 1189-1197.
- Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.
- U.S.EPS, "Guidelines for developing an air quality (Ozone and PM2.5) forecasting program", 2003, EPA-456/R-03-002.

DEVELOPING MACHINE LEARNING-BASED PREDICTIVE MODELS OF ADVERSE DRUG RESPONSES¹

Introduction

As medicine advances, drugs (medications) are increasingly used for disease prevention, alleviation, and treatment; accordingly, adverse drug responses (ADRs) are increasing as well. ADRs are a major cause of fatalities and illnesses, and are perceived not only as individuals' problems but also as a major social issue.

In the US, 6.7% of inpatients are estimated to experience ADRs during their hospital stay. In 1994, the estimated number of deaths in the US due to ADRs was approximately 100,000; it ranked as the fourth cause of death in the country, after heart disease (approximately 750,000 deaths annually), cancer (approximately 530,000), and strokes (approximately 150,000) (Lazarou et al., 1998). In the UK, 6.5% of all hospital admissions are related to ADRs, and, in 80% of them, ADRs are suggested as a direct factor (Pirmohamed et al., 2004). In the EU, it is reported that an average of 3.5% of hospital admissions are due to ADRs and that 10.1% of patients experience ADRs during their hospital stay (Bouvy et al., 2004).

Significant socioeconomic losses are also attributable to ADRs. A US study with outpatients reported that the cost incurred due to the prevalence of drug-related illnesses and fatalities has more than doubled between 1995 and 2000, from \$76.6 billion to \$177.4 billion. This means that the total cost due to side effects and deaths exceeded the total cost of the drugs themselves (Ernst et al., 2001). The UK estimates the annual hospital-stay cost related to ADRs to be £466 million, which is a significant economic loss (Pirmohamed et al., 2004). South Korea is experiencing high levels of drugs use, and thus, drugs side effects are expected to increase continuously (Park, 2016).

The drug-use level in South Korea ranks first worldwide; it is double the level in the US (Science Times, 2012) (<http://www.scientetimes.co.kr>).

¹ In this study, 'drugs Side Effects and the Risk of Narcotic Addiction' are determined as 'adverse drug responses'.

Moreover, with changes in social factors, e.g., aging and longer life spans, the number of people experiencing drugs side effects is predicted to increase (BioPharm, 2012)(<http://www.biopharminternational.com>). In South Korea, the average hospital stay is 3.9 days for patients experiencing ADRs during the stay, and the average healthcare payment is 930,420 won; these are 31.6% and 20.7% higher(Koo, 2008), respectively, than patients without such experiences. Thus, ADRs worsen patients' health and raise healthcare costs, ultimately adding to social costs.

Recently, the self-drugs market has become active, as over-the-counter (OTC) drugs have become available at convenience stores. This creates blind spots in safety management with respect to drug side effects, and increases the number of unlawful websites selling drugs, the spread of false drugs in the market, unapproved drug use, among others. ADRs are generally not detected, even though approximately 60%–70% of them are preventable (Greener, 2014; Cossey, 2010). Therefore, countries around the world have expended considerable interest and effort in pharmacovigilance to detect drug-related incidents early, improve public health and safety issues, and reduce socioeconomic losses. In particular, problems involving drugs are always immediately preceded by abnormal signs or precursors²; if they are not detected early, the problem increases due to a failure to deal with the changes in a timely manner, as well as judgment errors.

Pre-marketing clinical trials, which are conducted to evaluate drug efficacy and safety, are limited in detecting ADRs because of small sample sizes, relatively short study periods, and a lack of diversity in study participants (e.g., the exclusion of pediatric or elderly patients, patients taking other drugs, among others)(Sultana et al., 2013). To overcome the pre-marketing surveillance drawbacks, protect patients from post-marketing safety issues, and minimize fatalities to patients, a large-scale population study must be conducted to perform continuous drug-safety surveillance and monitor potential ADRs.

In the past several decades, post-marketing surveillance has been conducted using spontaneous reporting systems. A spontaneous reporting system is generally supported by a government regulatory agency, e.g., the US Food and Drug Administration (FDA) or the European Medicines Agency (EMA). Pharmaceutical companies, healthcare providers, and patients can directly report cases when ADRs are suspected. Spontaneous reports by patients are known to be more detailed and contain more

² According to Heinrich's law, before a major incident occurs, 29 minor warning-like incidents will occur and prior to that, 300 signs will manifest (the law of 1:29:300).

temporal information than those provided by healthcare providers, thereby increasing the likelihood of detecting previously unknown ADRs (Aagaard et al., 2009; Avery et al., 2011; Van et al., 2007; Vilhelmsson et al., 2011). However, the effectiveness of spontaneous reporting systems is limited by the underreporting of ADRs. Hazell and Shakir (2006) estimated that more than 90% of ADRs are unreported. Additionally, it is difficult to accurately assess the actual risk level of a particular drug in patients because clinical information is often insufficient and data on the denominator population treated with the drug are unavailable (Chung et al., 2012).

At present, big data is receiving considerable interest as an important active-surveillance data source to complement the limitations of voluntary reporting systems and allow the meaningful investigation of safety issues. In addition, the rapid growth of health-related information that can be processed via computer, together with the technological advances that made it possible to automatically process an enormous amount of data using natural language processing (NLP) and machine learning algorithms, opened up more opportunities for detecting and predicting ADRs using big data, e.g., electronic medical records (EMRs) and social media data.

Indeed, active drug-monitoring projects have been conducted worldwide in the last few years, e.g., the Sentinel Initiative in the US, the EU-ADR Project in the EU, and MIHARI (the Medical Information for Risk Assessment Initiative) in Japan. These projects use large-scale computerized data, e.g., health-insurance claims data and EMRs, to detect and demonstrate drug side effects. In South Korea, proposals have been made to establish a system to monitor drug-safety incidents using the big data available at the Korean National Health Insurance Services (Park, 2016).

Using a standardized and systematic terminology to report ADRs is crucial because it enables the information to be communicated accurately and safety measures to be utilized. Accordingly, many efforts have been made to standardize ADR terminology. The global ADR terminology systems used at the present time are WHO-ART (World Health Organization Adverse Reaction Terminology) developed by WHO in 1969 and MedDRA (Medical Dictionary for Regulatory Activities) developed by ICH (the International Council for Harmonisation of Technical Requirements for Pharmaceutical for Human Use) in 1994 (KIDS, 2014). Currently, an integrated system of international standard terminology, based on WHO-ART and MedDRA, is being actively pursued to meet the needs of those involved in drug surveillance (Fig. 1: Uppsala Monitoring Centre, 2017).

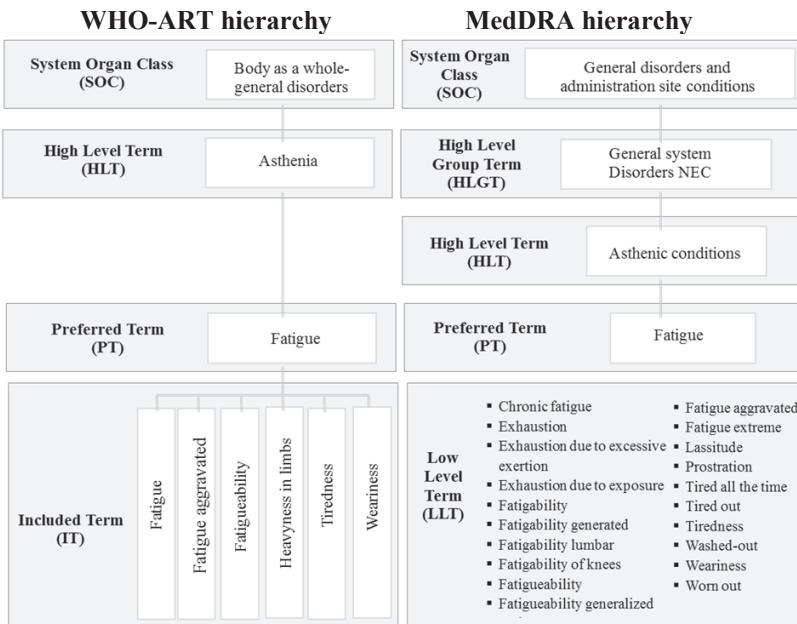


Fig. 1. WHO-ART & MedDRA hierarchy example (PT: Fatigue)

Narcotic addiction problems in South Korea have, thus far, not been as serious as in some other countries, including the US. However, the user base has recently been expanding to include even housewives and students. Thus, a systematic effort is being made to prevent the problem from spreading further. The United Nations Office on Drugs and Crime (UNODC) estimated that the number of individuals worldwide between ages 15 and 64 who used illegal drugs at least once in 2009 were 153–300 million (3.4%–6.6% of the world population). Approximately 12% of them, i.e., 3.86–15.5 million, were classified as "problem drug(narcotic) users" who have a narcotic addiction or a narcotic-abuse disorder (Supreme Prosecutors' Office of the Republic of Korea, 2015).

Recently, narcotic-related crimes have been increasing in South Korea, as narcotic trafficking via the Internet and social network service (SNS) has occurred, and means for individuals to buy illegal drugs have increased (Supreme Prosecutors' Office of the Republic of Korea, 2015: p. 97). According to the 2015 analysis, the number of narcotic-related crimes in South Korea was 11,916, the highest thus far. This number was higher than the previous year's number (9,984 in 2014) by 19.4% (Supreme

Prosecutors' Office of the Republic of Korea, 2015). Accordingly, the government announced that a system to constantly monitor the Internet would be immediately developed and operated to shut down unlawful websites, and the relevant information would be aggressively investigated.

In South Korea, narcotics are defined as substances that cause physical/mental dependencies and affect the central nervous system to excessively excite or inhibit the functions. They should be regulated according to the relevant laws. In general, narcotic drugs are classified into (1) excitors (stimulants) and inhibitors (relaxants), based on the pharmacological mechanism; (2) addictive and habitual drugs from the perspective of dependency; (3) natural and synthetic / semisynthetic drugs in terms of the origin; and (4) opiates, psychotropic drugs, and cannabis, according to the manufacturer (Supreme Prosecutors' Office of the Republic of Korea, 2015).

In terms of origin, narcotic drugs are categorized as follows (Korea Association against Drug Abuse, 2017):

- Natural opiates (opium, morphine, heroin, and cocaine),
- Synthetic opiates (methadone and pethidine hydrochloride),
- Psychotropic drugs (methamphetamines (philopon), barbiturates, benzodiazepines, Lysergic acid diethylamide (LSD), and mescaline),
- Cannabis, and
- Inhalants (glue and gases).

According to their pharmacological mechanism, they are categorized as follows:

- Stimulants (amphetamines, cocaine, methamphetamine, methylphenidate, and nicotine),
- Hallucinogens (LSD, mescaline, phencyclidine, psilocybin, amphetamine analogs, cannabis, hashish, tetrahydrocannabinol, ketamine, and anabolic),
- Opium and morphine (codeine, heroin, methadone, morphine, opium, and oxycodone), and
- Sedative drug (alcohol, barbiturates, benzodiazepines, GHB, and methoqualone).

Finally, according to their type, they are categorized as follows:

- Natural opiates (opium alkaloid preparations and coca alkaloid preparations),

- Synthetic opiates,
- Psychotropic drugs,
- Inhalants, and
- Excitors and inhibitors of the central nervous system

The influence of social media is expanding with the exponential growth of its data. If social media characteristics, e.g., an enormous amount of data and almost instantaneous information exchanges, are utilized well, it may be possible to monitor ADRs in real time and provide potential opportunities to discover many more ADR incidents and promptly detect early signals.

Currently, the value of reports on safety issues, made not only by healthcare professionals but also patients themselves, is greatly recognized in detecting unknown ADRs(Blenkinsopp et al., 2006; Hughes & Cohen, 2011). Hence, considerable attention is being paid to drug surveillance by real patients on widely used social-networking sites, e.g., Twitter (Twitter, 2017), generally, and health-related sites, e.g., PatientsLikeMe (PatientsLikeMe, 2017), DailyStrength (DailyStrength, 2017), and MedHelp (MedHelp, 2017), more specifically. Research is actively being conducted in this area (Sarker et al., 2015; Lardon et al., 2015; Sloane et al., 2015).

Recent studies have suggested combining existing drug-safety data with the big data spontaneously generated by patients on social media as an appropriate tool to further facilitate post-marketing drug surveillance. Although in the recent past, many studies have been carried out to detect and predict ADRs using social media, most of them were conducted outside of South Korea on texts written in English. Hence, the present study aims to collect drug-related texts from all accessible online channels, use machine learning to analyze drug-related documents (buzz) contained in social media data, and develop predictive models for drugs side effects and narcotic addiction risk.

Research Subjects and Analysis Method

Research Materials

The research materials of the present study were social big data collected from South Korean online news sites, blogs, Internet cafes, social network services, and online bulletin boards. Specifically, social big data was defined in the present study as text-based web documents (buzz) collected through a total of 280 online channels, which included 257

online news sites, four blog sites (Naver, Tistory, Daum, and Egloos), two Internet cafes (Naver and Daum), one SNS (Twitter), and 16 online bulletin boards (Naver Knowledge, Nate Knowledge, Nate Talk, Nate Pann, among others).

Drug-related topics were collected on an hourly basis every day from January 1, 2014 through April 30, 2017, including weekends and holidays. A total of 2,786,441 texts (575,447 in 2014; 711,195 in 2015; 1,113,862 in 2016; and 385,937 in 2017) were collected for analysis. Social big data was collected using a web crawler and the topics were classified using a text-mining method. To obtain all relevant texts, "medication," "narcotic," "drug," among others, were used as topics and topic synonyms. To eliminate noise, hair perm products and homophones to "drug" in the Korean language (e.g., Old Testament and New Testament, dog bedding marketed as a "Narcotic-like Cushion," Tsushima Island, among others) were used as stop words.

Research Tools

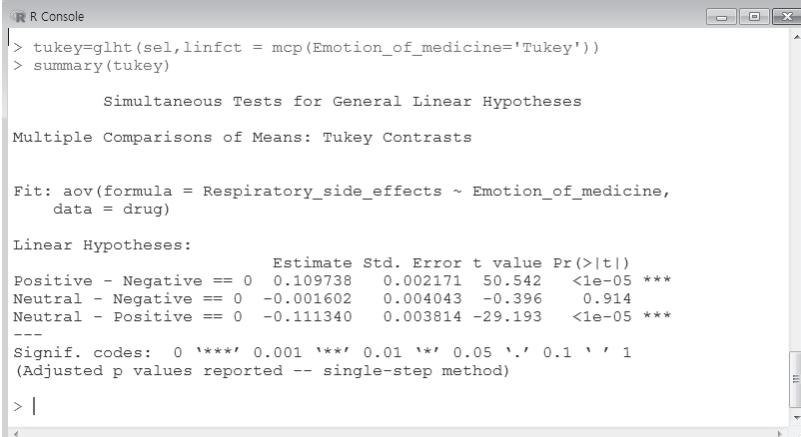
The social media data were processed using text mining and then numeric-coded for making structured data.

1) Sentiment analysis of adverse drug responses(ADRs)

Sentiment analysis of ADRs(positive, negative), i.e., a dependent variable of the study, was defined through theme analysis on emotional keywords. "Possible"–"healing" were defined as positive and "fake"–"sacrifice" as negative. Positive sentiment was coded 1 and negative sentiment coded 3. If both positive and negative sentiments were equally present in the document, it was determined to be neutral and coded 2. A positive sentiment toward drugs meant a positive feeling that the drug did not have a certain respiratory side effect (symptom), and a negative sentiment meant a negative feeling that it had the side effect (symptom).

In contrast, positive sentiment toward narcotics meant a dangerous feeling of liking them, and a negative sentiment meant a general feeling of disapproval. After analysis of variance (ANOVA) was conducted, the final sentiment toward medications was categorized into positive and negative (negative + neutral), as shown in Table 1. The final sentiment toward narcotics was categorized into dangerous (positive) and general (neutral + negative).

Table 1 ANOVA analysis adverse drug responses (ADRs) sentiment



The screenshot shows an R console window titled "R Console". The code entered is:

```
> tukey=glht(sel, linfct = mcp(Emotion_of_medicine='Tukey'))
> summary(tukey)
```

The output is:

```
Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = Respiratory_side_effects ~ Emotion_of_medicine,
         data = drug)

Linear Hypotheses:
Estimate Std. Error t value Pr(>|t|)
Positive - Negative == 0 0.109738 0.002171 50.542 <1e-05 ***
Neutral - Negative == 0 -0.001602 0.004043 -0.396 0.914
Neutral - Positive == 0 -0.111340 0.003814 -29.193 <1e-05 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

> |

Interpretation: After ANOVA was performed with respiratory symptoms as the dependent variable, the sentiment toward medications (positive, neutral, and negative) was categorized into two groups, i.e., positive (coded 1) and negative (neutral + negative; coded 0).

2) Respiratory symptoms (side effects)

To predict the risk of medication side effects, theme analysis was performed on respiratory symptoms (side effects). The following nine symptoms were defined as respiratory symptoms: bronchial sore throat, choking, coughing, lung disease, difficulty in breathing, sneezing, asthma, mucus, and sputum. Each of these symptoms was coded "n" if present, and "0" if not(absent).

3) Narcotic

The following 19 substances were defined as narcotics through thematic analysis: Opium, morphine, heroin, cocaine, codeine, amphetamine, benzodiazepines, LSD, cannabis, marijuana, propofol, precursor chemicals, ecstasy, stimulants, psychotropic drugs, hallucinogens, and new drug. Each of these narcotics was coded as "n" if it was present and "0" if not.

Statistical Analysis

The following methodology was used to construct the most efficient predictive models to explain drugs side effects and the risk of narcotic addiction. Drugs and narcotic-related online texts were collected using a web crawler; then, the keywords were classified via text mining and opinion mining (sentiment analysis). After the classified keywords were numerically coded, future signals were searched using term frequency and document frequency. Once future signals were found, machine learning analysis was conducted to discover new phenomena and make predictions via a classification process. The search for the main narcotic signals was performed by computing the degree of visibility (DoV) and degree of diffusion (DoD) and confirming with a keyword emergence map (KEM) and keyword issue map (KIM). Predictive modeling and visualization were performed using various machine learning analytic techniques.

To explore future signs by analyzing documents in the form of text collected from on-line channels, first, the term frequency (TF) and document frequency (DF) in the collected documents should be extracted by performing text mining. TF can be extracted by calculating the frequency of emergence of each term in each document and then combining the frequency of emergence in each document. DF refers to the number of documents in which a certain term appears. For extraction of important information from text mining, the method of Term Frequency - Inverse Document Frequency (TF-IDF) is used. TF-IDF is a statistical figure that expresses the importance of a specific term in the document in question out of a group of documents (Jung, 2010). Spärck (1972) suggested Inverse Document Frequency [$IDF_j = \log_{10}(\frac{N}{DF_j})$] for the purpose of assigning greater weights to rare terms. Accordingly, in a case when a greater weight needs to be assigned to a rarer term in the analysis of term frequency, the term frequency and inverse document frequency are combined to calculate ' $TF-IDF = TF_{ij} \times IDF_j$ ' and to apply a proper weight (index of a term's importance).

Recently, various studies have been conducted to detect environmental changes in the future and the one that draws the greatest attention among them involves detection of weak signals that help people foresee future changes (Yoon, 2012). Weak signals are signs of latent (or potential) change in the future (Ansoff, 1975). Weak signals can evolve into strong signals, and then strong signals can grow into trends or mega-trends over time. Hiltunen (2008) explained the weak signal using a three-dimensional spatial concept including signal, issue, and interpretation, by using the

concept of future sign. Yoon (2012) collected web-news documents and connected the term frequency and document frequency generated from text-mining analysis to Hiltunen's (2008) signal and issue, respectively. Yoon (2012) used the term frequency, document frequency, and rate increase of emergence frequency to develop the key-word portfolio for Keyword Emergence Map (KEM) and Keyword Issue Map (KIM) and then extracted out weak signals by using the developed key-word portfolio. KEM displays visibility and calculates the Degree of Visibility (DoV), while KIM shows the degree of diffusion and calculates the Degree of Diffusion (DoD).

$$DoV_{ij} = \left(\frac{TF_{ij}}{NN_j} \right) \times \{1 - tw \times (n-j)\} \quad (1)$$

$$DoD_{ij} = \left(\frac{DF_{ij}}{NN_j} \right) \times \{1 - tw \times (n-j)\} \quad (2)$$

In our discussions, NN represents the total number of documents; TF , term frequency; DF , document frequency; tw , time weight (for this study, a time weight of 0.05 was applied); n , the entire time segment; and j , a time point. ' $\{1 - tw \times (n-j)\}$ ' is a function that weakens its influence as time goes by, which can determine the scale of the time weight. Park and Kim(2015) evaluated future signs that can be discovered in IoT news in the energy field as in the manner demonstrated in Fig. 2 based on the research methodology proposed by Hiltunen (2008) and Yoon (2012).

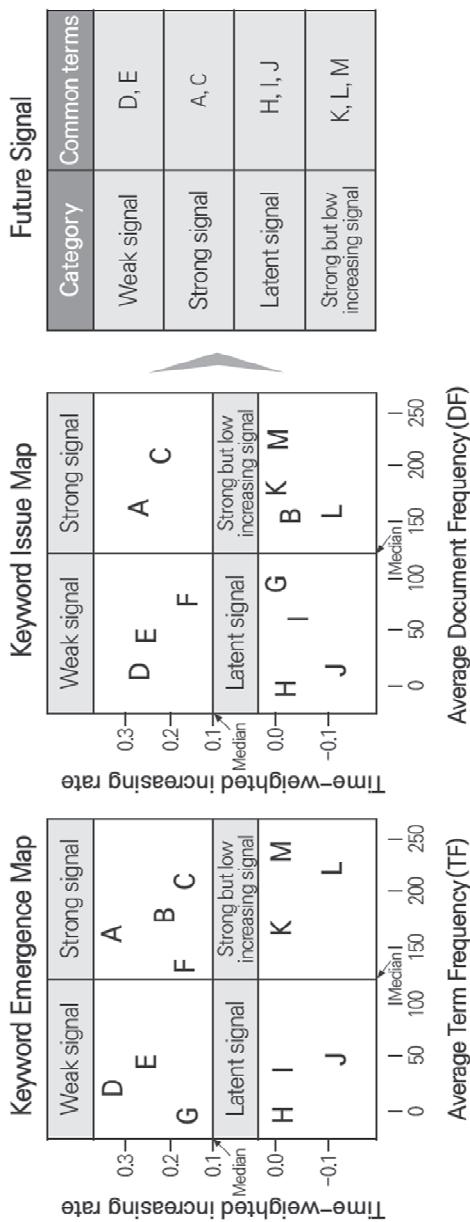


Fig. 2. Future signal derivation process

Result

Current Status of ADRs-related Online Documents

The current statuses of drugs and narcotic-related online documents are shown in Table 2. Sentiment toward medications consisted of positive (64.2%), neutral (7.1%), and negative (28.7%) feelings. The most common respiratory side effect was coughing (25.1%), followed by asthma (20.5%) and sputum (12.8%). The most frequently occurring narcotic was cannabis (18.8%), followed by hallucinogens (17.9%), benzodiazepines (9.9%), new narcotic analogs (8.5%), and stimulants (7.5%).

Table 2 Online document of drugs side effects and narcotics

Item	Variables	N(%)	Item	Variables	N(%)	
Emotion	Positive	995,370(64.2)	Narcotics	Opium	4,582(1.9)	
	Neutral	110,681(7.1)		Morphine	4,113(1.7)	
	Negative	444,620(28.7)		Heroin	4,357(1.8)	
	Total	1,550,671		Cocaine	11,575(4.8)	
Respiratory side effects	Bronchial sore throat	1,816(2.0)		Codeine	2,651(1.1)	
	Choke	6,547(7.2)		Amphetamine	7,266(3.0)	
	Cough	22,799(25.1)		Benzodiazepines	24,191(9.9)	
	Lung disease	9,681(10.7)		Lysergic acid diethylamide	1,747(0.7)	
	Difficulty in breathing	7,644(8.4)		Cannabis	45,758(18.8)	
	Sneeze	4,085(4.5)		Marijuana	7,727(3.2)	
	Asthma	18,563(20.5)		Propofol	16,319(6.7)	
	Snot	8,003(8.8)		Precursor chemical	7,371(3.0)	
	Sputum	11,582(12.8)		Estasy	6,271(2.6)	
	Total	90,720		Stimulant	18,329(7.5)	
				Psychotropic Drugs	1,6531(6.8)	
				Hallucinogenics	43,573(17.9)	
				New drug	20,773(8.5)	
				Total	243,134	

Narcotic-related Future Signals

Table 3 shows the changes in signals for narcotics obtained through TF-IDF analysis, which considers the term frequency (TF), inverse document frequency (IDF), and importance index.

In term frequency, cannabis ranked first, followed by hallucinogens, new narcotic analogs, benzodiazepines, propofol, stimulants, and psychotropic drugs. In document frequency, cannabis ranked first again, followed by hallucinogens, benzodiazepines, stimulants, new narcotic analogs, propofol, and psychotropic drugs. In particular, new narcotic analogs ranked fifth in document frequency, which is indicative of their diffusion level; however, they ranked second in term frequency, which takes the importance index into account. Therefore, plans to monitor and manage new narcotic analogs should be established because they can develop into a threat.

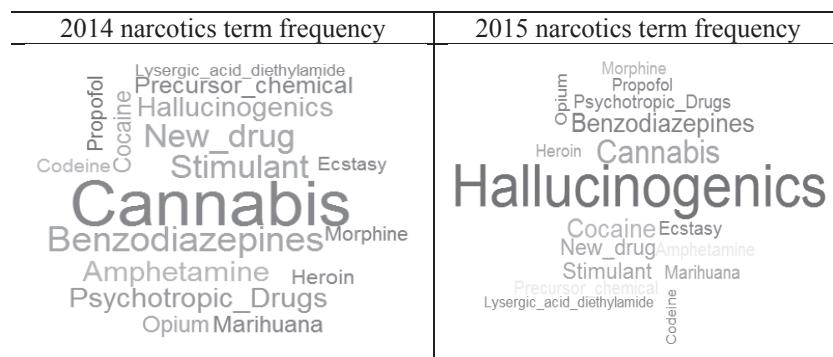
Table 3 Keyword analysis of narcotics in online channels

Ranking	Term frequency		Document frequency		Term frequency-inverse document frequency	
	Keyword	Frequency	Keyword	Frequency	Keyword	Frequency
1	Cannabis	56419	Cannabis	40689	Cannabis	40262
2	Hallucinogenics	36136	Hallucinogenics	34350	New_drug	37808
3	New_drug	34034	Benzodiazepines	21511	Hallucinogenics	28445
4	Benzodiazepines	24510	Stimulant	16711	Propofol	25266
5	Propofol	22128	New_drug	16301	Benzodiazepines	24276
6	Stimulant	19910	Propofol	15180	Stimulant	21903
7	Psychotropic_Drugs	17241	Psychotropic_Drugs	14175	Psychotropic_Drugs	20199
8	Cocaine	14575	Cocaine	10936	Amphetamine	19753
9	Amphetamine	13398	Amphetamine	7059	Cocaine	18718
10	Precursor_chemical	12045	Precursor_chemical	6819	Precursor_chemical	17940
11	Marijuana	9928	Marijuana	5720	Marijuana	15544
12	Opium	6680	Ecstasy	4763	Opium	11310
13	Ecstasy	5852	Opium	4265	Ecstasy	9628
14	Heroin	5553	Heroin	4070	Heroin	9515
15	Morphine	5440	Morphine	3789	Morphine	9490
16	Codeine	3224	Codeine	2497	Codeine	6208
17	Lysergic_acid_diethylamide	2521	Lysergic_acid_diethylamide	1589	Lysergic_acid_diethylamide	5349
	Total	289594	Total	210424	Total	321615

Changes in yearly narcotic rankings are shown in Table 4. Hallucinogens ranked tenth in 2014, but they were first in 2015, fifth in 2016, and first again until April 2017. Accordingly, a system to monitor hallucinogens should be established (Fig. 3).

Table 4 Annual keyword ranking of narcotics in online channels (TF)

Ranking	2014Year	2015Year	2016Year
1	Cannabis	Hallucinogenics	Cannabis
2	New_drug	Cannabis	Propofol
3	Amphetamine	New_drug	New_drug
4	Benzodiazepines	Cocaine	Benzodiazepines
5	Stimulant	Benzodiazepines	Psychotropic_Drugs
6	Psychotropic_Drugs	Stimulant	Hallucinogenics
7	Propofol	Precursor_chemical	Stimulant
8	Marihuana	Propofol	Precursor_chemical
9	Precursor_chemical	Psychotropic_Drugs	Cocaine
10	Hallucinogenics	Ecstasy	Marihuana
11	Cocaine	Marihuana	Amphetamine
12	Opium	Amphetamine	Opium
13	Heroin	Opium	Morphine
14	Morphine	Morphine	Heroin
15	Ecstasy	Heroin	Ecstasy
16	Codeine	Lysergic_acid_diethylamide	Codeine
17	Lysergic_acid_diethylamide	Codeine	Lysergic_acid_diethylamide



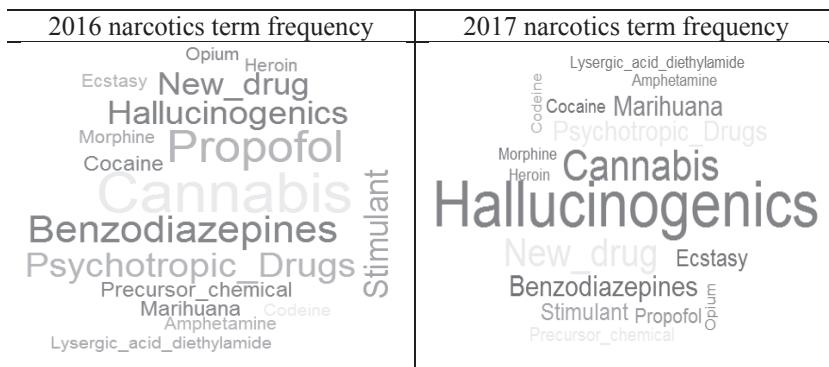


Fig. 3. Narcotics visualization (wordcloud)

The DoV increase rate in narcotics and the mean term frequency were computed. The results showed that the DoV increased with a rate of 0.128 and the frequency of the narcotic-related keywords had increased. Particularly, hallucinogens, benzodiazepines, and propofol appeared quite frequently, and their increase rates were higher than the median value. Therefore, systems for monitoring and managing these narcotics should be established. The DoD showed a trend similar to DoV, but the diffusion speeds of propofol and hallucinogens were rapid (Table 5 and 6).

Table 5 DoV mean increase rate and mean term frequency(TF) for narcotics

Keyword	DoV			Mean increase rate	Mean term frequency
	2014 Year	2015 Year	2016 Year		
Cannabis	19601	14977	21841	0.018	18806
Hallucinogenics	3402	24190	8544	2.265	12045
New_drug	10463	11254	12317	-0.017	11345
Benzodiazepines	6570	7262	10678	0.173	8170
Propofol	3836	3556	14736	1.352	7376
Stimulant	6154	5283	8473	0.128	6637
Psychotropic Drugs	4186	3317	9738	0.725	5747
Cocaine	2639	8391	3545	0.584	4858
Amphetamine	8430	2756	2212	-0.48	4466
Precursor_chemical	3582	4060	4403	0.004	4015
Marihuana	3645	2941	3342	-0.114	3309
Opium	2504	2090	2086	-0.167	2227
Ecstasy	1197	2997	1658	0.351	1951

Heroin	1990	1817	1746	-0.15	1851
Morphine	1500	1890	2050	0.059	1813
Codeine	900	749	1575	0.351	1075
Lysergic acid diethylamide	592	780	1149	0.267	840
Median				0.128	4466

Table 6 DoD mean increase rate and mean term frequency(TF) for narcotics

Keyword	DoD			Mean increase rate	Mean term frequency
	2014 Year	2015 Year	2016 Year		
Cannabis	13755	10191	16743	0.074	13563
Hallucinogenics	3006	23789	7555	2.5	11450
Benzodiazepines	5761	6434	9316	0.143	7170
Stimulant	5154	4350	7207	0.124	5570
New_drug	4794	4719	6788	0.082	5434
Propofol	1561	1512	12107	3.114	5060
Psychotropic_Drugs	3437	2749	7989	0.683	4725
Cocaine	1948	6568	2420	0.599	3645
Amphetamine	4010	1736	1313	-0.467	2353
Precursor_chemical	2122	2248	2449	-0.047	2273
Marihuana	1785	1682	2253	0.019	1907
Ecstasy	909	2633	1221	0.442	1588
Opium	1800	1215	1250	-0.238	1422
Heroin	1452	1275	1343	-0.141	1357
Morphine	1169	1216	1404	-0.025	1263
Codeine	777	568	1152	0.248	832
Lysergic_acid_diethylamide	372	364	853	0.498	530
Median				0.124	2353

As shown in (Table 7, Fig. 4, Fig. 5), strong signals in both KEM and KIM (the first quadrant) were observed in hallucinogens, propofol, psychotropic drugs, cocaine, benzodiazepines, and stimulants. Weak signals (the second quadrant) were observed in codeine, ecstasy, and LSD. ‘Strong but low increasing signal(the fourth quadrant)’ in KEM and KIM were found in cannabis and new narcotic analogs, while latent(potential) signals common in KEM and KIM in the third quadrant were found in morphine, heroin, opium, marijuana, precursor chemicals, and amphetamines. In particular, propofol, located in the first quadrant and associated with strong signals, showed a high increase rate. Accordingly, a system to manage propofol should be urgently established.

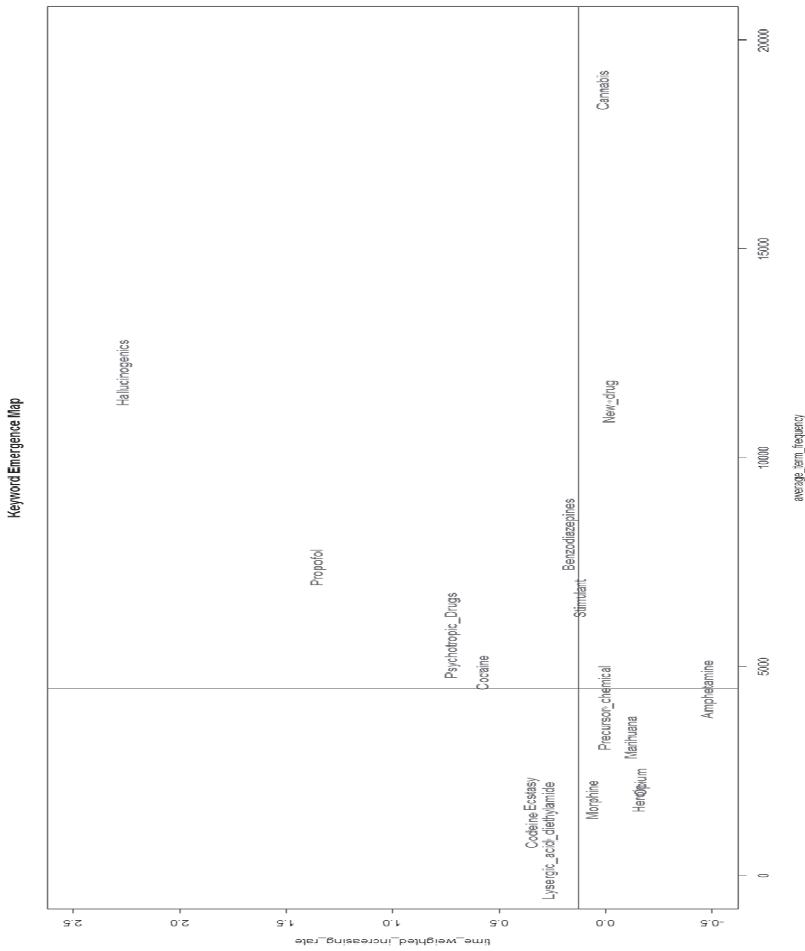
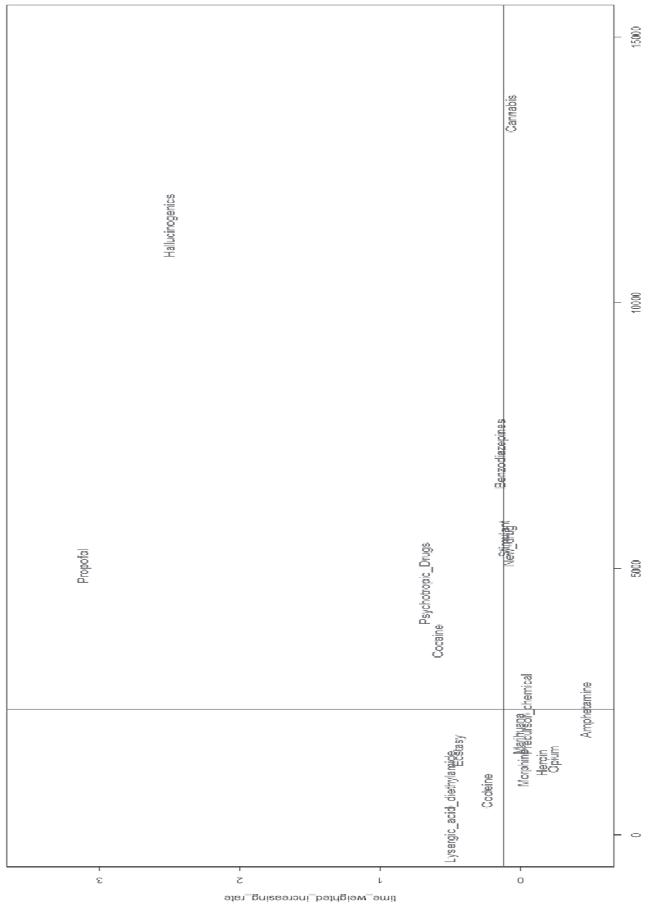


Fig. 4. Keyword emergence map for narcotic



```

R Console
> ## future signal Drugs DoD
>
> rm(list=ls())
> setwd("c:/drug_2017")
> data_spss=read.table(file="future_DoD_drug.txt",header=T)
> windows(height=8.5, width=8)
> plot(data_spss$tf,data_spss$df,xlim=c(0,15000), ylim=c(-.5,3.5), pch=18,
+ col=8,xlab='average_document_frequency', ylab='time_weighted_increasing_rate',
+ main='Keyword Issue Map')
> text(data_spss$tf,data_spss$df,label=data_spss$Drugs,cex=1.2, col='red')
> abline(h=0.124, v=2353, lty=1, col=4, lwd=0.5)
> savePlot('future_DoD_drug',type='pdf')
> |

```

Fig. 5. Keyword issue map for narcotic

Table 7 Future signals of narcotics -related keywords

Future signal	Latent signal	Weak signal	Strong signal	Strong but low increasing signal
KEM	Morphine, Heroin, Opium, Marihuana, Precursor_chemical, Amphetamine	Codeine, Ecstasy, Lysergic acid diethylamide	Hallucinogenics, Propofol, Psychotropic_Drugs, Cocaine, Benzodiazepines, Stimulant	Cannabis, New_drug
KIM	Morphine, Heroin, Opium, Marihuana, Precursor_chemical, Amphetamine	Codeine, Ecstasy, Lysergic acid diethylamide	Hallucinogenics, Propofol, Psychotropic_Drugs, Cocaine, Benzodiazepines, Stimulant	Cannabis, New_drug
Main signal	Morphine, Heroin, Opium, Marihuana, Precursor_chemical, Amphetamine	Codeine, Ecstasy, Lysergic acid diethylamide	Hallucinogenics, Propofol, Psychotropic_Drugs, Cocaine, Benzodiazepines, Stimulant	Cannabis, New_drug

Development of machine learning-based predictive models for drugs side effects and risk of narcotic addiction

1) Developing neural network predictive model

■ Respiratory symptoms (side effects)

Of the data containing the input variables' frequencies for respiratory symptoms, the 2014–2017 data (51,628 cases) were used as a learning dataset. To develop predictive models, training data and test data were sampled from the learning dataset with a ratio of 50:50. Nine respiratory symptoms (bronchoconstriction, choking, coughing, chronic obstructive pulmonary disease (COPD), difficulty in breathing, sneezing, asthma, mucus, and sputum) were submitted to a multilayer neural network with an input layer, five hidden layers, and an output layer. The results are presented in Fig. 6.

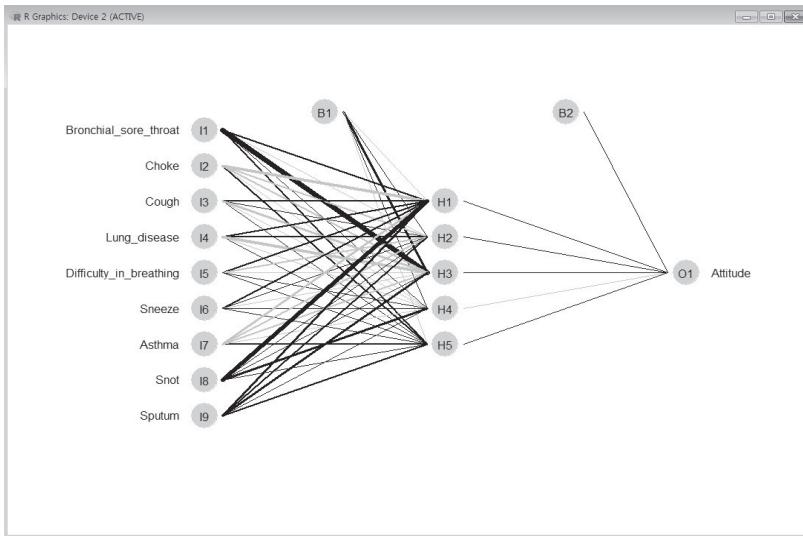
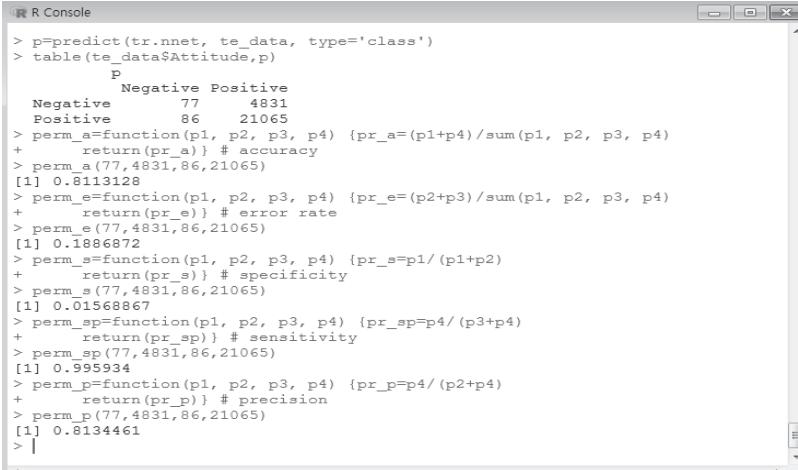


Fig. 6. Neural network predictive model (Respiratory symptoms)

The predictive model was evaluated in the following manner. Training data and test data were constructed by randomly sampling 50% of the entire dataset. A model function was developed, based on training data, and predictions were made on the test data. The model was evaluated using the error distribution by comparing the actual and predicted groups (classification groups). The accuracy of the neural-network predictive model for respiratory symptoms was 81.13%, with an error rate of 18.87%, sensitivity of 99.59%, specificity of 1.57%, and precision of 81.34% (Table 8).

Table 8 Evaluation of neural network predictive model (Respiratory symptoms)


```

> p=predict(tr.nnet, te_data, type='class')
> table(te_data$Attitude,p)
   P
Negative    77    4831
Positive   86   21065
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(77,4831,86,21065)
[1] 0.8113128
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(77,4831,86,21065)
[1] 0.1886872
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(77,4831,86,21065)
[1] 0.01568867
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)} # sensitivity
> perm_sp(77,4831,86,21065)
[1] 0.995934
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(77,4831,86,21065)
[1] 0.8134461
>

```

■ Risk of narcotic addiction

The narcotic neural network model for sentiment toward narcotics predicted general feelings with a mean probability of 22.92%, and dangerous feelings with a mean probability of 77.08% (Fig. 7).

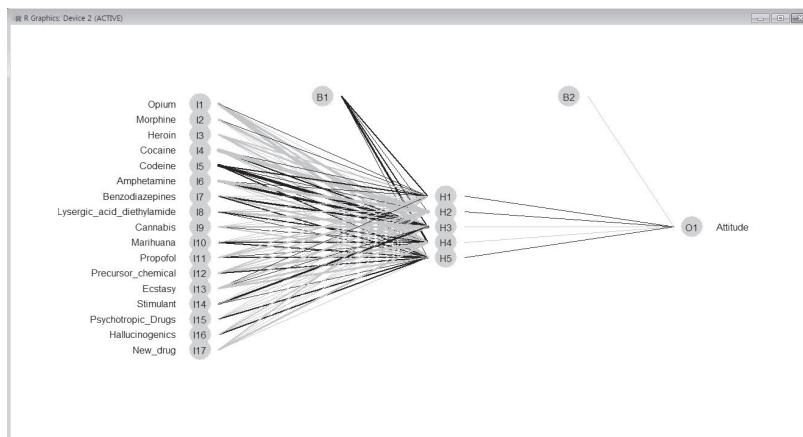


Fig. 7. Neural network predictive model(narcotic addiction)

2) Developing logistic predictive model

■ Respiratory symptoms (side effects)

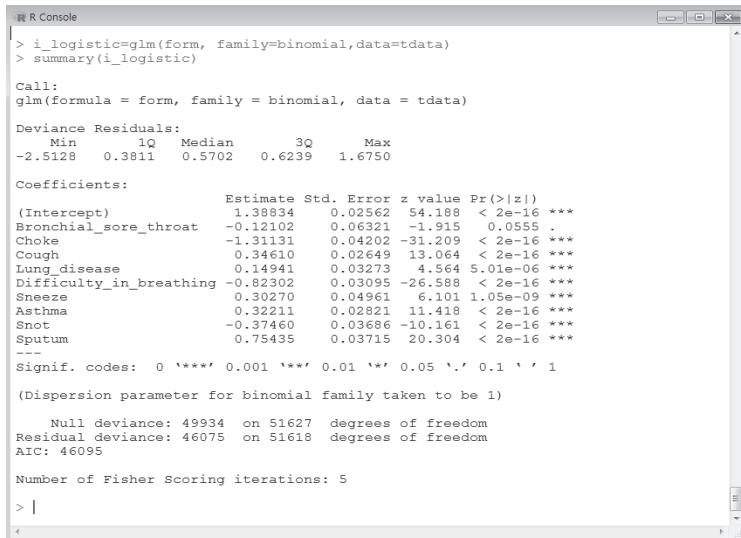
Fig. 8 shows the results of assessing the logistic predictive model for respiratory symptoms. The accuracy was 81.08%, error rate was 18.92%, sensitivity was 99.78%, specificity was 1.17%, and precision was 81.19%. Coughing, COPD, sneezing, asthma, and sputum had higher probabilities of presence, whereas bronchoconstriction, choking, difficulty in breathing, and mucus had higher probabilities of absence (Table 9).



The screenshot shows the R Console window with the following content:

```
|> tdata = read.table('Respiratory_N_20180527.txt',header=T)
|> input=read.table('input_Respiratory.txt',header=T,sep=",")
|> output=read.table('output_Respiratory.txt',header=T,sep=",")
Warning message:
In read.table("output_Respiratory.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Respiratory.txt'
> predict_data=read.table('p_output.txt',header=T)
Warning message:
In read.table("p_output.txt", header = T) :
  incomplete final line found by readTableHeader on 'p_output.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Bronchial_sore_throat + Choke + Cough + Lung_disease +
  Difficulty_in_breathing + Sneeze + Asthma + Snot + Sputum
> ind=sample(2, nrow(tdata), replace=T, prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> i_logistic=glm(form, family=binomial,data=tr_data)
> p=predict(i_logistic,te_data,type='response')
> p=round(p)
> table(te_data$Attitude,p)
  p
    0     1
  0   57 4827
  1   46 20832
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(57,4827,46,20832)
[1] 0.8108454
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(57,4827,46,20832)
[1] 0.1891546
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(57,4827,46,20832)
[1] 0.01167076
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)} # sensitivity
> perm_sp(57,4827,46,20832)
[1] 0.9977967
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(57,4827,46,20832)
[1] 0.8118789
> |
```

Fig. 8. Logistic predictive model (narcotic addiction)

Table 9 Logistic regression model (narcotic addiction)


```
R Console
> i_logistic<-glm(form, family=binomial,data=tdata)
> summary(i_logistic)

Call:
glm(formula = form, family = binomial, data = tdata)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-2.5128   0.3811   0.5702   0.6239   1.6750 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) 1.38834  0.02562 54.188 < 2e-16 ***
Bronchial_sore_throat -0.12102  0.06321 -1.915 0.0555 .  
Choke        -1.31131  0.04202 -31.209 < 2e-16 ***
Cough         0.34610  0.02649 13.064 < 2e-16 ***
Lung_disease 0.14941  0.03273  4.564 5.01e-06 ***
Difficulty_in_breathing -0.82302  0.03095 -26.588 < 2e-16 ***
Sneeze       0.30270  0.04961  6.101 1.05e-09 *** 
Asthma        0.32211  0.02821 11.418 < 2e-16 *** 
Snot          -0.37460  0.03686 -10.161 < 2e-16 *** 
Sputum        0.75435  0.03715 20.304 < 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: 49934  on 51627  degrees of freedom
Residual deviance: 46075  on 51618  degrees of freedom
AIC: 46095

Number of Fisher Scoring iterations: 5
```

■ Risk of narcotic addiction

The logistic regression model for sentiment toward narcotics predicted general feelings with a mean probability of 22.98%, and dangerous feelings with a mean probability of 77.02% (Fig. 9).



```
R Console
> ## logistic regression modeling
>
> rm(list=ls())
> setwd("c:/drug_2017")
> tdata = read.table('drug_N_20180527.txt',header=T)
> input=read.table('drug_input.txt',header=T,sep=",")
Warning message:
In read.table("drug_input.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'drug_input.txt'
> output=read.table('output_drug.txt',header=T,sep=",")
Warning message:
In read.table("output_drug.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_drug.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(output_vars, collapse = '+'), '-',
+ paste(input_vars, collapse = '+')) 
> form
Attitude ~ Opium + Morphine + Heroin + Cocaine + Codeine + Amphetamine +
  Benzodiazepines + Lysergic acid diethylamide + Cannabis +
  Marijuana + Propofol + Phenobarbital + Ecstasy + Stimulant +
  Psychotropic_Drugs + Hallucinogenics + New_drug
> i_logistic<-glm(form, family=binomial,data=tdata)
> p=predict(i_logistic,tdata,type='response')
> mean(p)
[1] 0.7701782
> |
```

Fig. 9. logistic predictive model (narcotic addiction)

3) Developing support vector machine predictive model

■ Respiratory symptoms (side effects)

The results assessing the support vector machine model for respiratory symptoms are shown in Fig. 10. The accuracy was 81.44%, error rate was 18.56%, sensitivity was 99.53%, specificity was 3.20%, and precision was 81.65%.



R Console window showing R code for developing an SVM model for respiratory symptoms. The code reads data from three files: 'Respiratory_S_20180527.txt', 'input_Respiratory.txt', and 'output_Respiratory.txt'. It then performs various operations like reading tables, creating variables, fitting an SVM model with a radial kernel, and calculating performance metrics (accuracy, error rate, sensitivity, specificity, and precision).

```

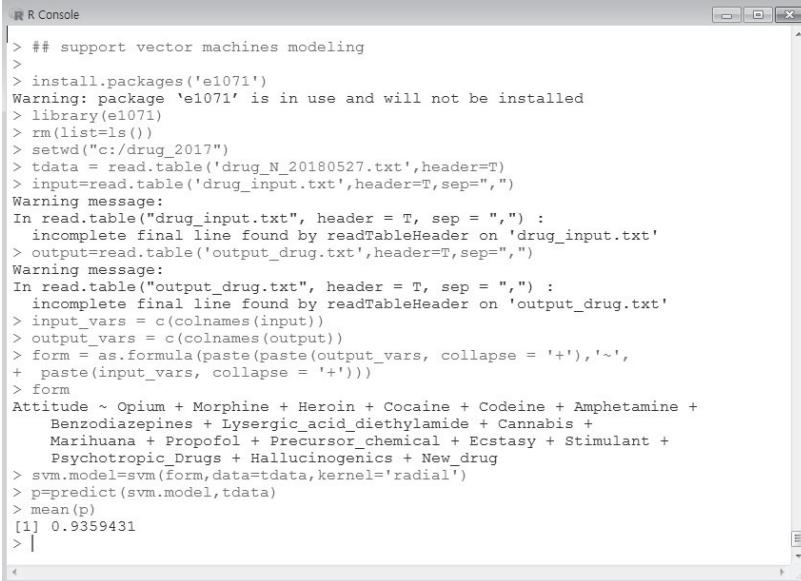
> tdata = read.table('Respiratory_S_20180527.txt',header=T)
> input=read.table('input_Respiratory.txt',header=T,sep=",")
> output=read.table('output_Respiratory.txt',header=T,sep=",")
Warning message:
In read.table("output_Respiratory.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Respiratory.txt'
> predict_data=read.table('p_output.txt',header=T)
Warning message:
In read.table("p_output.txt", header = T) :
  incomplete final line found by readTableHeader on 'p_output.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Bronchial_sore_throat + Choke + Cough + Lung_disease +
  Difficulty_in_breathing + Sneeze + Asthma + Snot + Sputum
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> svm.model=svm(form,data=tr_data,kernel='radial')
> #summary(svm.model)
> p=predict(svm.model,te_data)
> table(te_data$Attitude,p)
   p
     Negative Positive
Negative      155     4696
Positive       99    20889
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(155,4696,99,20889)
[1] 0.8144278
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(155,4696,99,20889)
[1] 0.1855722
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(155,4696,99,20889)
[1] 0.03195217
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)} # sensitivity
> perm_sp(155,4696,99,20889)
[1] 0.995283
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(155,4696,99,20889)
[1] 0.816455
> |

```

Fig. 10. support vector machine predictive model (Respiratory symptoms)

■ Risk of narcotic addiction

The support vector machine model for sentiment toward narcotics predicted general feelings with a mean probability of 6.41% and dangerous feelings with a mean probability of 93.59% (Fig. 11).



```
R Console
| > ## support vector machines modeling
| >
| > install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
| > library(e1071)
| > rm(list=ls())
| > setwd("c:/drug_2017")
| > tdata = read.table('drug_N_20180527.txt',header=T)
| > input=read.table('drug_input.txt',header=T,sep=",")
Warning message:
In read.table("drug_input.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'drug_input.txt'
| > output=read.table('output_drug.txt',header=T,sep=",")
Warning message:
In read.table("output_drug.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_drug.txt'
| > input_vars = c(colnames(input))
| > output_vars = c(colnames(output))
| > form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
| > form
Attitude ~ Opium + Morphine + Heroin + Cocaine + Codeine + Amphetamine +
  Benzodiazepines + Lysergic_acid_diethylamide + Cannabis +
  Marijuana + Propofol + Precursor_chemical + Ecstasy + Stimulant +
  Psychotropic_Drugs + Hallucinogenics + New_drug
| > svm.model=svm(form,data=tdata,kernel='radial')
| > p=predict(svm.model,tdata)
| > mean(p)
[1] 0.9359431
| >
```

Fig. 11. support vector machine predictive model (narcotic addiction)

4) Developing random forest predictive model

■ Respiratory symptoms (side effects)

The results assessing the random forest model for respiratory symptoms are shown in Fig. 12. The accuracy was 81.63%, error rate was 18.37%, sensitivity was 99.68%, specificity was 2.28%, and precision was 81.76%.

The importance plot of the random forest model for respiratory symptoms revealed that choking, difficulty in breathing, sputum, coughing, asthma, and mucus, in that order, were the most influential for the sentiment toward medications (Fig. 13).



The screenshot shows an R console window titled "R Console". The code in the console is as follows:

```

R Console

> tdata = read.table('Respiratory_S_20180527.txt',header=T)
> input=read.table('input_Respiratory.txt',header=T,sep=",")
> output=read.table('output_Respiratory.txt',header=T,sep=",")
Warning message:
In read.table("output_Respiratory.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Respiratory.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Bronchial_sore_throat + Choke + Cough + Lung_disease +
  Difficulty_in_breathing + Sneeze + Asthma + Snot + Sputum
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> tdata_rf = randomForest(form, data=tr_data ,forest=FALSE,importance=TRUE)
> p=predict(tdata_rf,te_data)
> table(te_data$Attitude,p)
      p
      Negative Positive
Negative      110     4717
Positive       68    21149
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(110,4717,68,21149)
[1] 0.8162725
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(110,4717,68,21149)
[1] 0.1837275
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(110,4717,68,21149)
[1] 0.02278848
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)} # sensitivity
> perm_sp(110,4717,68,21149)
[1] 0.996795
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(110,4717,68,21149)
[1] 0.8176371
> |

```

Fig. 12. Random forest predictive model (Respiratory symptoms)

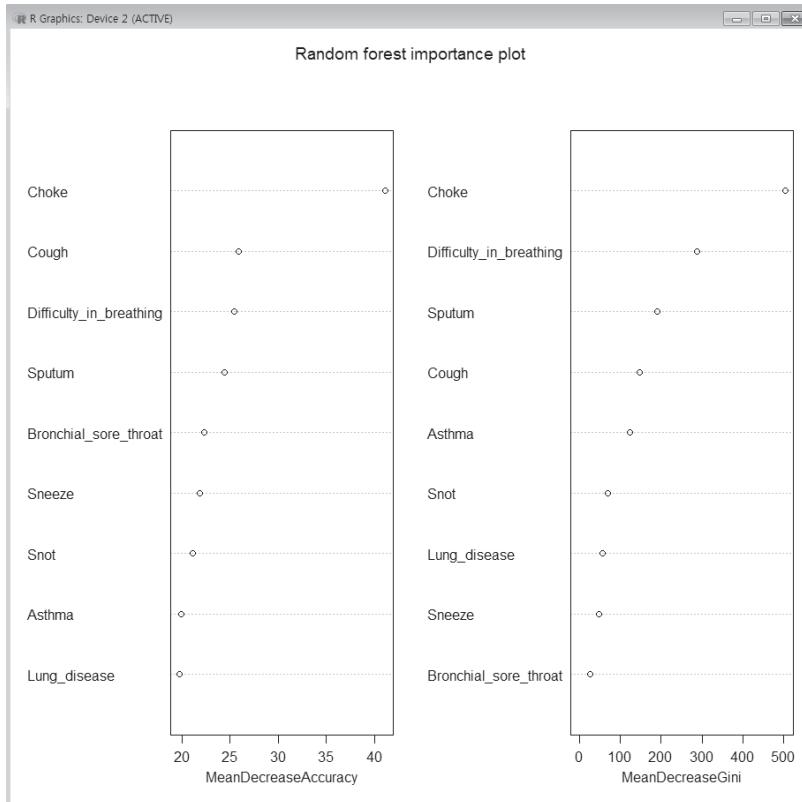
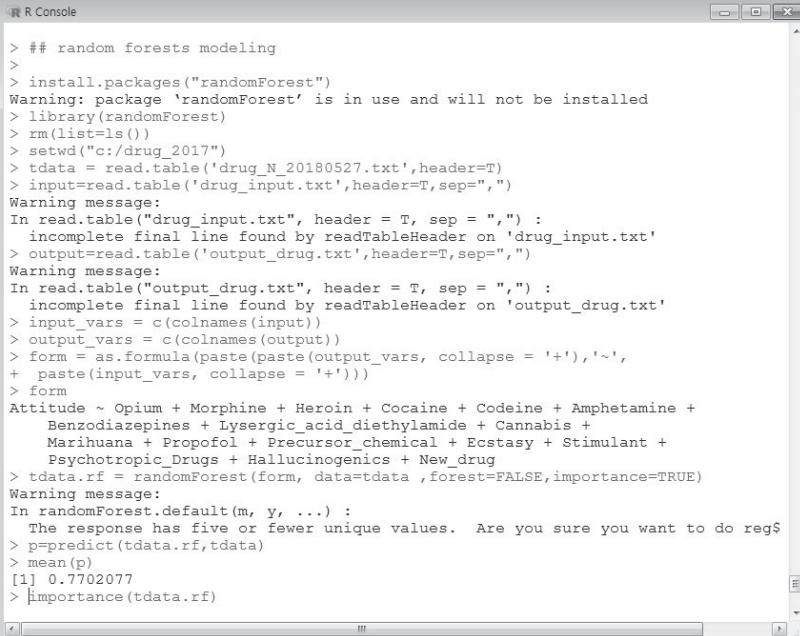


Fig. 13. Importance plot of random forest model (Respiratory symptoms)

■ Risk of narcotic addiction

The random forest model for sentiment toward narcotics predicted general feelings with a mean probability of 22.98% and dangerous feelings with a mean probability of 77.02% (Fig. 14).



The screenshot shows the R Console window with the following R script:

```
> ## random forests modeling
>
> install.packages("randomForest")
Warning: package 'randomForest' is in use and will not be installed
> library(randomForest)
> rm(list=ls())
> setwd("c:/drug_2017")
> tdata = read.table('drug_N_20180527.txt',header=T)
> input=read.table('drug_input.txt',header=T,sep=",")
Warning message:
In read.table("drug_input.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'drug_input.txt'
> output=read.table('output_drug.txt',header=T,sep=",")
Warning message:
In read.table("output_drug.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_drug.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Opium + Morphine + Heroin + Cocaine + Codeine + Amphetamine +
  Benzodiazepines + Lysergic_acid_diethylamide + Cannabis +
  Marijuana + Propofol + Precursor_chemical + Ecstasy + Stimulant +
  Psychotropic_Drugs + Hallucinogenics + New_drug
> tdata.rf = randomForest(form, data=tdata ,forest=FALSE,importance=TRUE)
Warning message:
In randomForest.default(m, y, ...):
  The response has five or fewer unique values. Are you sure you want to do reg$
```

The console output shows the creation of a random forest model named `tdata.rf` using the `randomForest` package. The model is trained on the `tdata` dataset with `Attitude` as the outcome variable and various drug categories as predictors. The importance of each predictor is calculated.

Fig. 14. Random forest predictive model (narcotic addiction)

According to the random forest importance plot (`IncNodePurity`), the narcotics most influential to sentiment toward narcotics (general, dangerous) were the new narcotic analogs, followed by cannabis, amphetamines, propofol, cocaine, heroin, and hallucinogens (Fig. 15).

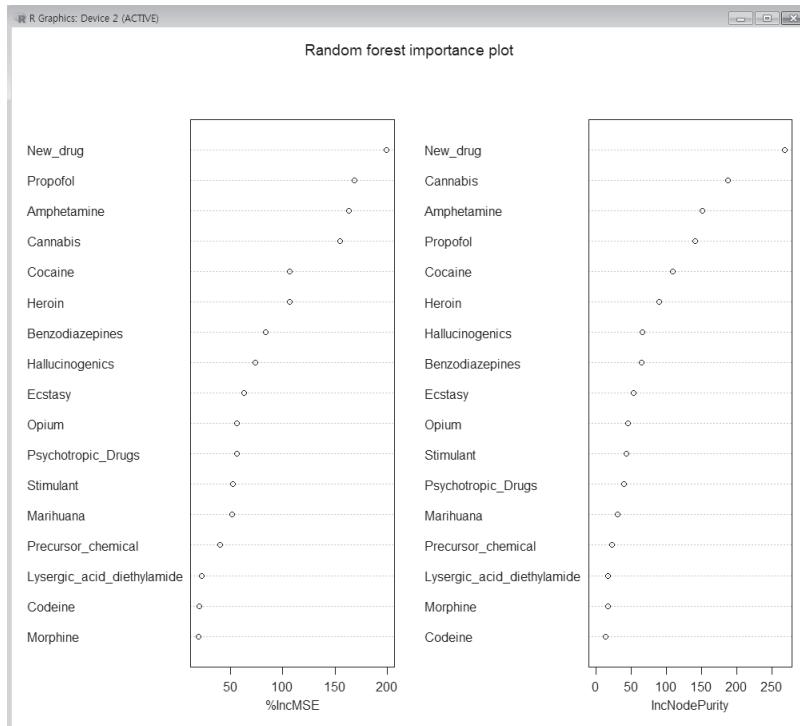


Fig. 15. Importance plot of random forest model (narcotic addiction)

5) Developing decision tree model

■ Respiratory symptoms (side effects)

The results assessing the decision tree model for respiratory symptoms are shown in Fig. 16. The accuracy was 81.25%, error rate was 18.75%, sensitivity was 99.60%, specificity was 1.94%, and precision was 81.44%.



```

R Console

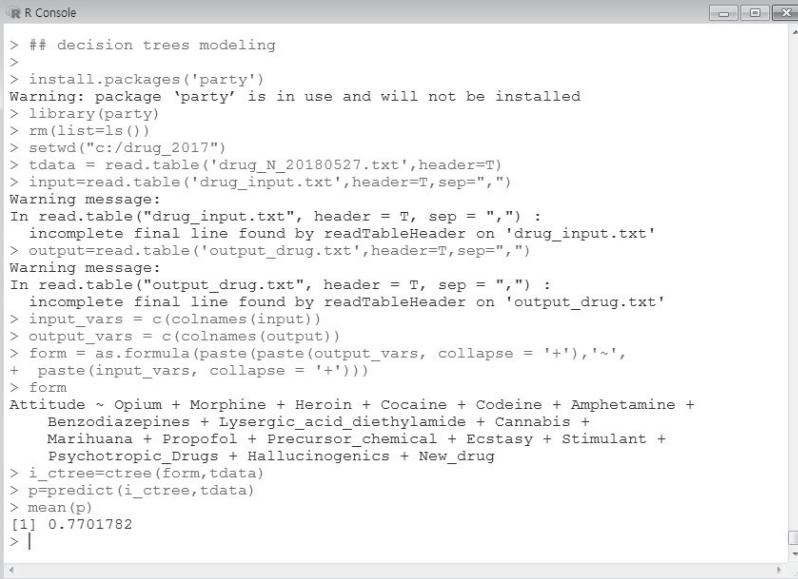
> tdata = read.table('Respiratory_S_20180527.txt',header=T)
> input=read.table('input_Respiratory.txt',header=T,sep=",")
> output=read.table('output_Respiratory.txt',header=T,sep=",")
Warning message:
In read.table("output_Respiratory.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Respiratory.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Bronchial_sore_throat + Choke + Cough + Lung_disease +
  Difficulty_in_breathing + Sneeze + Asthma + Snot + Sputum
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> i_ctree=ctree(form,tr_data)
> p=predict(i_ctree,te_data)
> table(te_data$Attitude,p)
      p
      Negative Positive
Negative      94     4750
Positive       84    20847
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(94,4750,84,20847)
[1] 0.8124539
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(94,4750,84,20847)
[1] 0.1875461
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(94,4750,84,20847)
[1] 0.01940545
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)} # sensitivity
> perm_sp(94,4750,84,20847)
[1] 0.9959868
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(94,4750,84,20847)
[1] 0.8144314
>

```

Fig. 16. Decision tree model (Respiratory symptoms)

■ Risk of narcotic addiction

The random forest The decision tree model for sentiment toward narcotics predicted general feelings with a mean probability of 22.98% and dangerous feelings with a mean probability of 77.02%(Fig. 17).



The screenshot shows the R Console window with the following R script:

```
> ## decision trees modeling
>
> install.packages('party')
Warning: package 'party' is in use and will not be installed
> library(party)
> rm(list=ls())
> setwd("c:/drug_2017")
> tdata = read.table('drug_N_20180527.txt',header=T)
> input=read.table('drug_input.txt',header=T,sep=",")
Warning message:
In read.table("drug_input.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'drug_input.txt'
> output=read.table('output_drug.txt',header=T,sep=",")
Warning message:
In read.table("output_drug.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_drug.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Opium + Morphine + Heroin + Cocaine + Codeine + Amphetamine +
  Benzodiazepines + Lysergic_acid_diethylamide + Cannabis +
  Marihuana + Propofol + Precursor_chemical + Ecstasy + Stimulant +
  Psychotropic Drugs + Hallucinogenics + New_drug
> i_ctree=ctree(form,tdata)
> p=predict(i_ctree,tdata)
> mean(p)
[1] 0.7701782
> |
```

Fig. 17. Decision tree model (narcotic addiction)

New narcotic analogs were the most influential in the sentiment toward narcotics, followed by cannabis (Fig. 18).

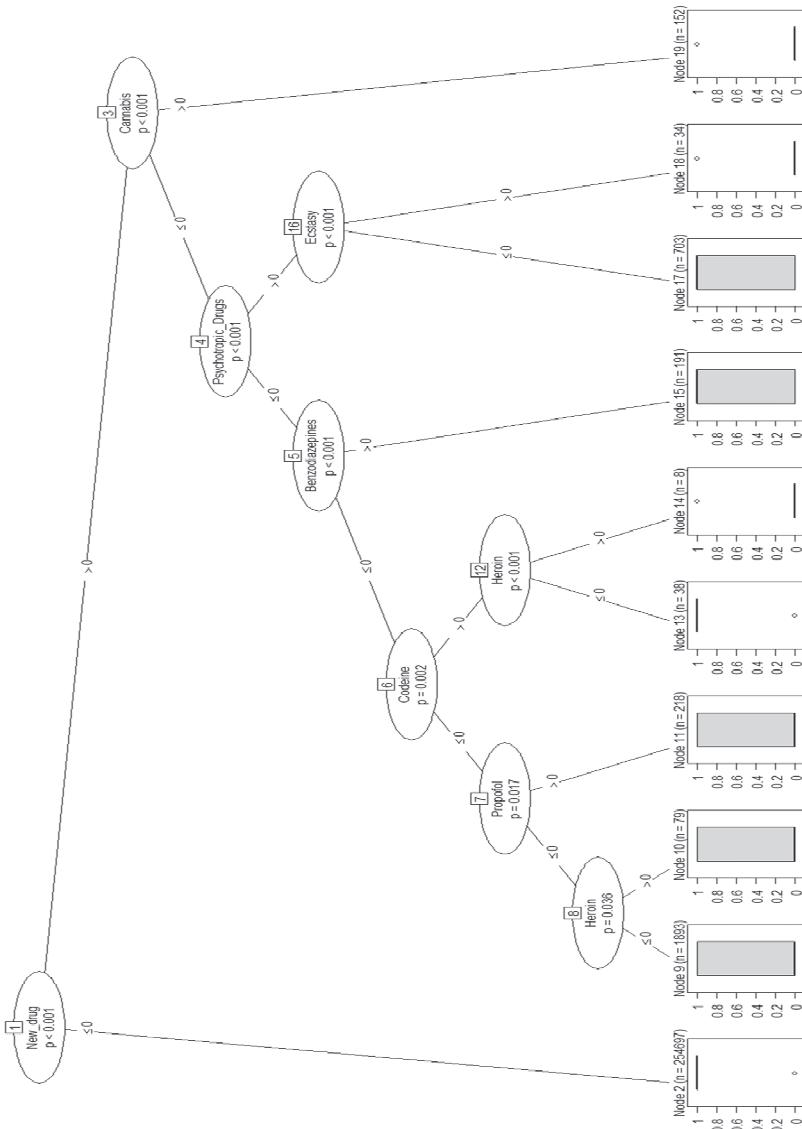


Fig. 18. Decision tree model plot (narcotic addiction)

6) Developing Naïve Bayesian classification model

■ Respiratory symptoms (side effects)

The results assessing the Naïve Bayesian classification model for respiratory symptoms are shown in Fig. 19. The accuracy was 78.14%, error rate was 21.87%, sensitivity was 88.10%, specificity was 36.07%, and precision was 85.33%.



```
R Console
> tdata = read.table('Respiratory_S_20180527.txt',header=T)
> input=read.table('input_Respiratory.txt',header=T,sep=",")
> output=read.table('output_Respiratory.txt',header=T,sep=",")
Warning message:
In read.table("output_Respiratory.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_Respiratory.txt'
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+'))))
> form
Attitude ~ Bronchial_sore_throat + Choke + Cough + Lung_disease +
  Difficulty_in_breathing + Sneeze + Asthma + Snot + Sputum
> ind=sample(2, nrow(tdata), replace=T,prob=c(0.5,0.5))
> tr_data=tdata[ind==1,]
> te_data=tdata[ind==2,]
> train_data.lda=naiveBayes(form,data=tr_data)
> ldapred=predict(train_data.lda, te_data, type='class')
> table(te_data$Attitude,ldapred)
ldapred
      Negative Positive
Negative     1782     3158
Positive      2480    18364
> perm_a=function(p1, p2, p3, p4) {pr_a=(p1+p4)/sum(p1, p2, p3, p4)
+   return(pr_a)} # accuracy
> perm_a(1782,3158,2480,18364)
[1] 0.7813373
> perm_e=function(p1, p2, p3, p4) {pr_e=(p2+p3)/sum(p1, p2, p3, p4)
+   return(pr_e)} # error rate
> perm_e(1782,3158,2480,18364)
[1] 0.2186627
> perm_s=function(p1, p2, p3, p4) {pr_s=p1/(p1+p2)
+   return(pr_s)} # specificity
> perm_s(1782,3158,2480,18364)
[1] 0.3607287
> perm_sp=function(p1, p2, p3, p4) {pr_sp=p4/(p3+p4)
+   return(pr_sp)} # sensitivity
> perm_sp(1782,3158,2480,18364)
[1] 0.8810209
> perm_p=function(p1, p2, p3, p4) {pr_p=p4/(p2+p4)
+   return(pr_p)} # precision
> perm_p(1782,3158,2480,18364)
[1] 0.8532664
> |
```

Fig. 19. Naïve Bayesian classification model (Respiratory symptoms)

The Naïve Bayesian classification model for sentiment toward narcotics predicted general feelings with a mean probability of 5.72% and dangerous feelings with a mean probability of 94.28% (Fig. 20).



R Console

```
> ## naiveBayes classification modeling
>
> install.packages('e1071')
Warning: package 'e1071' is in use and will not be installed
> library(e1071)
> rm(list=ls())
> setwd("c:/drug_2017")
>
> tdata = read.table('drug_N_20180527.txt',header=T)
> input=read.table('drug_input.txt',header=T,sep=",")
Warning message:
In read.table("drug_input.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'drug_input.txt'
> output=read.table('output_drug.txt',header=T,sep=",")
Warning message:
In read.table("output_drug.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'output_drug.txt'
> p_output=read.table('p_output_bayes.txt',header=T,sep=",")
Warning message:
In read.table("p_output_bayes.txt", header = T, sep = ",") :
  incomplete final line found by readTableHeader on 'p_output_bayes.txt'
>
> input_vars = c(colnames(input))
> output_vars = c(colnames(output))
> p_output_vars = c(colnames(p_output))
>
> form = as.formula(paste(paste(output_vars, collapse = '+'), '~',
+ paste(input_vars, collapse = '+')))
> form
Attitude ~ Opium + Morphine + Heroin + Cocaine + Codeine + Amphetamine +
  Benzodiazepines + Lysergic_acid_diethylamide + Cannabis +
  Marijuana + Propofol + Precursor_chemical + Ecstasy + Stimulant +
  Psychotropic_Drugs + Hallucinogenics + New_drug
>
> train_data.lda=naiveBayes(form,data=tdata, laplace=1)
> p=predict(train_data.lda, tdata, type='raw')
>
> dimnames(p)=list(NULL,c(p_output_vars))
> pred_obs = cbind(tdata, p)
> write.matrix(pred_obs,'drug_attitude_naive.txt')
> m_data = read.table('drug_attitude_naive.txt',header=T)
> #attach(m_data)
> mean(m_data$posterior.0)
[1] 0.05717015
> mean(m_data$posterior.1)
[1] 0.9428298
> |
```

Fig. 20. Naïve Bayesian classification model (narcotic addiction)

Evaluation of Machine Learning-based Predictive Models

1) Predictive models for respiratory symptoms

The machine learning models for respiratory symptoms were evaluated via ROC curves. The best predictive models were the random forest model, the neural network model, and the decision tree model, in that order (Fig. 21).

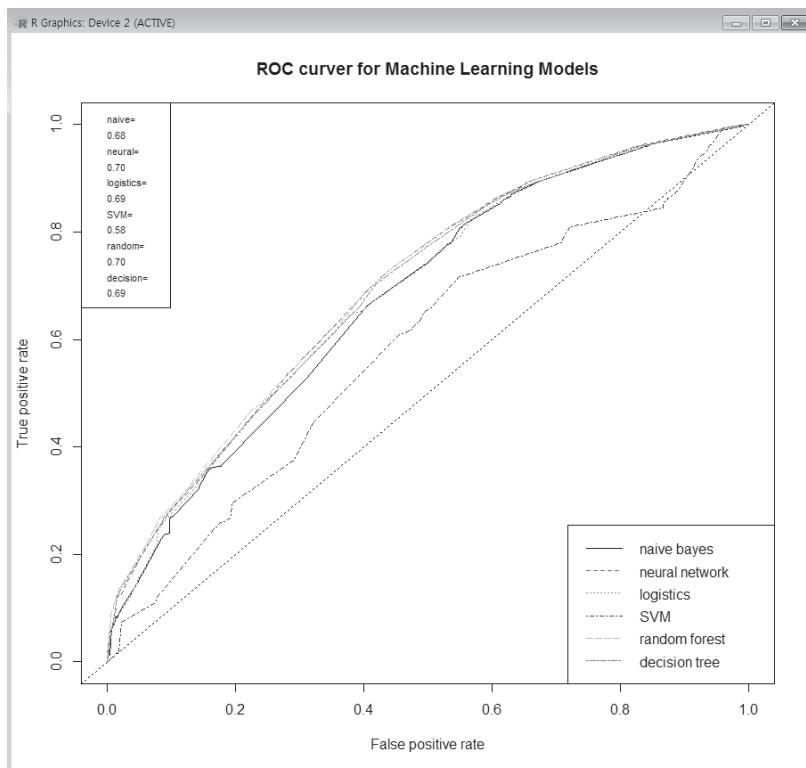


Fig. 21. Evaluation of machine learning(Respiratory symptoms)

2) Predictive models for the risk of narcotics

The ROC curve assessing the machine learning models for the risk of narcotic addiction showed that the models were not very good(Fig. 22).

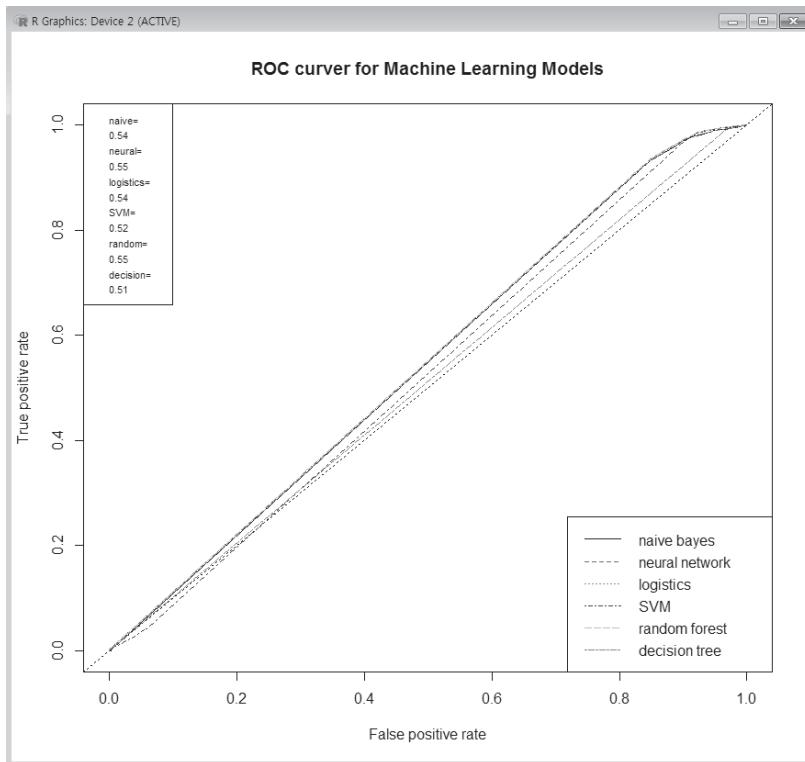


Fig. 22. Evaluation of machine learning(narcotic addiction)

Association Analysis

1) Respiratory symptoms

Regarding the association rules among the respiratory symptoms, the association of the four variables, {Cough, Sneeze, Sputum} => {Snot} had a support of 0.01, a confidence of 86.70, and a lift of 5.91. That is, a mention of "Cough, Sneeze, Sputum" side effects in an online document increases the probability for "Snot" to be mentioned by 5.91 times (Table 10).

Table 10 Association rules (Respiratory symptoms)

```

> rules.sorted=sort(rules1, by="lift")
> inspect(rules.sorted)
  lhs                                rhs      support    confidence   lift   count
[1] {Cough, Sneeze, Sputum}          => {Snot}  0.01022701  0.8669951  5.9051744  527
[2] {Sneeze, Sputum}                => {Snot}  0.01139917  0.8583942  5.8465928  588
[3] {Asthma, Snot}                  => {Sneaze} 0.01090494  0.4340786  5.6350546  563
[4] {Sneeze, Asthma}                => {Snot}  0.01090494  0.7929577  5.4009001  563
[5] {Sneeze}                        => {Snot}  0.05218099  0.6773950  4.6137929  2694
[6] {Cough, Sneeze}                 => {Snot}  0.02066708  0.6404562  4.3621994  1067
[7] {Cough, Asthma, Snot}           => {Sputum} 0.01020764  0.5317861  2.4300807  527
[8] {Asthma, Snot, Sputum}         => {Cough}  0.01020764  0.9705343  2.2755101  527
[9] {Cough, Sneeze, Snot}           => {Sputum} 0.01022701  0.4948454  2.2612742  528
[10] {Lung_disease, Asthma, Sputum}=> {Cough}  0.01433331  0.9487179  2.2243601  740
[11] {Difficulty_in Breathing, Sputum}=> {Cough}  0.01443015  0.9074300  2.1275565  745
[12] {Sneeze, Snot, Sputum}         => {Cough}  0.01022701  0.8979592  2.1053514  528
[13] {Snot, Sputum}                => {Cough}  0.02729139  0.8940355  2.0961520  1409
[14] {Sneeze, Sputum}              => {Cough}  0.01179592  0.8890511  2.0844655  609
[15] {Cough, Asthma}               => {Sputum} 0.06955528  0.4428413  2.0236334  3591
[16] {Asthma, Sputum}              => {Cough}  0.06955528  0.8505446  1.9941837  3591
[17] {Lung_disease, Sputum}        => {Cough}  0.02750445  0.8186561  1.9145061  1420
[18] {Asthma, Snot}                => {Sputum} 0.01051755  0.4186584  1.9131260  543
[19] {Cough, Snot}                 => {Sputum} 0.02729139  0.4053514  1.8523155  1409
[20] {Asthma, Snot}                => {Cough}  0.01919501  0.7640703  1.7914375  991
[21] {Lung_disease, Asthma}        => {Cough}  0.03970714  0.7008547  1.6432210  2050
[22] {Sputum}                      => {Cough}  0.14472767  0.6613560  1.5506125  7472
[23] {Cough, Lung_disease, Sputum}=> {Asthma} 0.01433331  0.5211268  1.4743127  740
[24] {Cough, Lung_disease}         => {Asthma} 0.03970714  0.5207010  1.4731083  2050
[25] {Difficulty_in Breathing, Asthma}=> {Cough}  0.01762609  0.6232877  1.4613577  910
[26] {Lung_disease, Difficulty_in Breathing}=> {Cough}  0.01365538  0.5939343  1.3925359  705
[27] {Cough, Sputum}               => {Asthma} 0.06955528  0.4805942  1.3596426  3591
[28] {Bronchial_sore_throat}       => {Cough}  0.01981483  0.5689655  1.3339942  1023
[29] {Lung_disease, Sputum}        => {Asthma} 0.01510808  0.4485333  1.2689405  780
[30] {Cough, Difficulty_in Breathing}=> {Asthma} 0.01762609  0.4469548  1.2644738  910
[31] {Snot}                        => {Cough}  0.06732781  0.4585752  1.0751735  3476
[32] {Asthma}                      => {Cough}  0.15706593  0.4443531  1.0418285  8109
[33] {}                            => {Cough}  0.42651275  0.4265127  1.0000000  22020
[34] {Sneeze}                      => {Cough}  0.032626931 0.4189087  0.9821716  1666
[35] {Lung_disease}                => {Cough}  0.07625707  0.4132899  0.9689979  3937
> |

```

The visualization of the association rules for respiratory symptoms shows that most respiratory symptoms are interlinked with Cough, Asthma, and Sputum (Fig. 23).

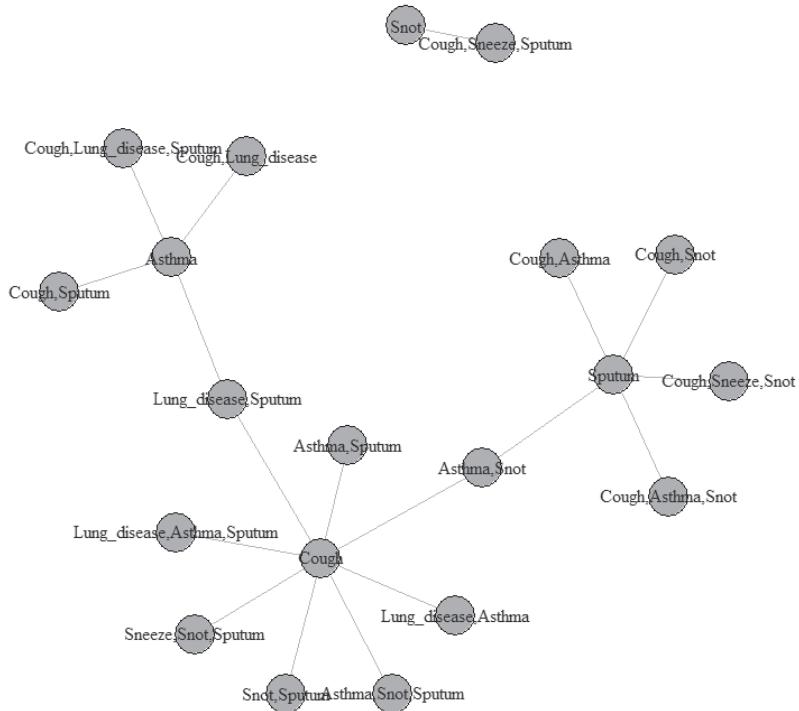
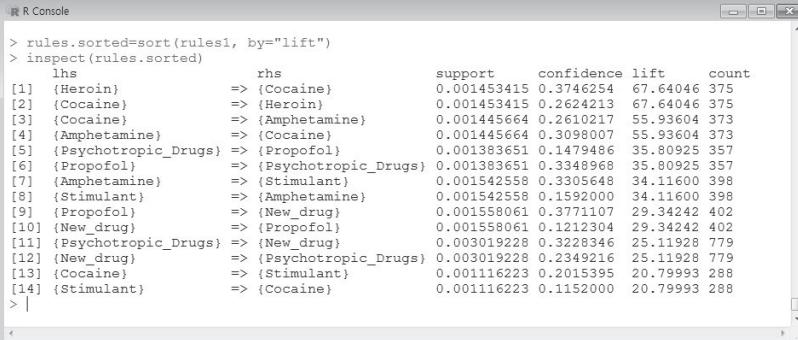


Fig. 23. Visualization of the association rules (Respiratory symptoms)

2) Narcotic addiction

The association prediction among narcotic-related keywords showed that the association of two variables, $\{\text{Heroin}\} \Rightarrow \{\text{Cocaine}\}$ had a support of 0.001, a confidence of 0.374, and a lift of 67.64. Thus, when 'Heroin' is mentioned in an online document, the probability of 'Cocaine' appearing is 37.4%, and the probability is 67.6 times higher, compared to when heroin is not mentioned (Table 11).

Table 11 Association rules (Narcotic addiction)


```
R Console

> rules.sorted=sort(rules1, by="lift")
> inspect(rules.sorted)
      lhs                  rhs          support    confidence   lift      count
[1] {Heroin}        => {Cocaine}    0.001453415  0.3746254 67.64046 375
[2] {Cocaine}       => {Heroin}    0.001453415  0.2624213 67.64046 375
[3] {Cocaine}       => {Amphetamine} 0.001445664  0.2610217 55.93604 373
[4] {Amphetamine}   => {Cocaine}    0.001445664  0.3098007 55.93604 373
[5] {Psychotropic_Drugs} => {Propofol} 0.001383651  0.1479486 35.80925 357
[6] {Propofol}       => {Psychotropic_Drugs} 0.001383651  0.3348968 35.80925 357
[7] {Amphetamine@}  => {Stimulant}  0.001542558  0.3305648 34.11600 398
[8] {Stimulant}     => {Amphetamine} 0.001542558  0.1592000 34.11600 398
[9] {Propofol}       => {New_drug}    0.001558061  0.3771107 29.34242 402
[10] {New_drug}      => {Propofol}   0.001558061  0.1212304 29.34242 402
[11] {Psychotropic_Drugs} => {New_drug}  0.003019228  0.3228346 25.11920 779
[12] {New_drug}      => {Psychotropic_Drugs} 0.003019228  0.2349216 25.11920 779
[13] {Cocaine}        => {Stimulant}  0.001116223  0.2015395 20.79993 288
[14] {Stimulant}     => {Cocaine}    0.001116223  0.1152000 20.79993 288
> |
```

Cluster Analysis

1) Respiratory symptoms

Cluster analysis is an analytic technique to classify individuals into a few homogeneous clusters, based on similarity. Prior to performing the cluster analysis, the number of clusters should be designated. On the scree plot of respiratory symptoms in the present study, the slope of the curve increased at six clusters. Therefore, the number of clusters was determined to be five (Fig. 24).

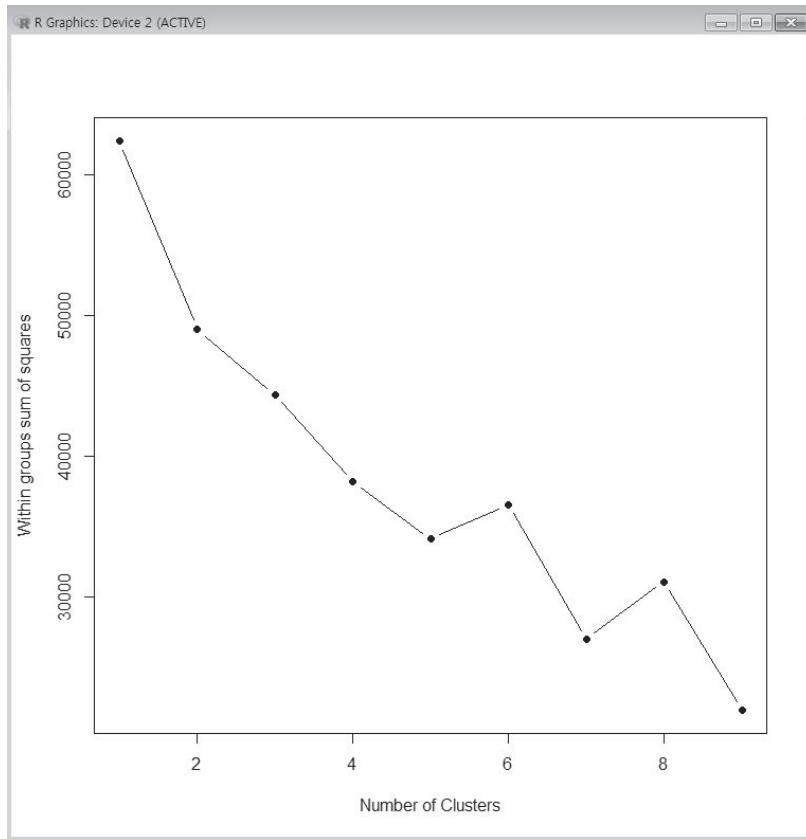


Fig. 24. Scree plot of cluster analysis (Respiratory symptoms)

After cluster analysis was used to extract five clusters, factors with a cluster mean of 0.3 or higher were included in each cluster. The final results were as follows. Cluster 1 had 21,201 cases and was labeled as "Cough", Cluster 2 had 6,374 cases and was labeled as "Difficulty in breathing", Cluster 3 had 6,803 cases and was labeled as "Cough, Sneeze, Snot", Cluster 4 had 11,890 cases and was labeled as "Asthma" Finally, cluster 5 had 5,360 cases and was labeled as "Cough, Lung disease, Asthma, and Sputum" (Table 12).

Table 12 Cluster analysis (Respiratory symptoms)

```
R Console

> fit = kmeans(clust_data, 5) # 5 cluster solution
> fit
K-means clustering with 5 clusters of sizes 21201, 6374, 6803, 11890, 5360

Cluster means:
  Bronchial_sore_throat Choke      Cough Lung_disease Difficulty_in_breathing
1          0.035281355 0.141408424 0.4917693    0.25739352           9.433517e-05
2          0.024474427 0.026513963 0.1386884    0.11123313          1.000000e+00
3          0.065706306 0.006320741 0.4211377    0.06232544           2.866382e-02
4          0.006896552 0.004205214 0.2121110    0.06156434          0.000000e+00
5          0.068097015 0.005037313 0.9930970    0.41119403           1.856343e-01

  Sneeze   Asthma     Sntr     Sputum
1 0.046743078 0.00000000 0.000000000 0.27177963
2 0.022434892 0.08346407 0.011295890 0.02714151
3 0.365427017 0.08849037 1.000000000 0.15125680
4 0.007737595 1.00000000 0.007905803 0.04802355
5 0.049440299 0.97481343 0.113992537 0.70205224
```

A detailed analysis was conducted to identify the clusters influential to sentiment toward medications (negative and positive). Negative sentiment toward respiratory side effects was the highest in cluster 1 (Cough: 42.33%), followed by cluster 2 (Difficulty in breathing: 22.45%) (Table 13).

Table 13 Segmentation analysis of cluster (Respiratory symptoms)

> ctab(t1, type=c("n", "r", "c", "t"))		Attitude	0	1
fit.cluster				
1	Count	4113.00	17088.00	
	Row %	19.40	80.60	
	Column %	42.33	40.77	
	Total %	7.97	33.10	
2	Count	2181.00	4193.00	
	Row %	34.22	65.78	
	Column %	22.45	10.00	
	Total %	4.22	8.12	
3	Count	1363.00	5440.00	
	Row %	20.04	79.96	
	Column %	14.03	12.98	
	Total %	2.64	10.54	
4	Count	1736.00	10154.00	
	Row %	14.60	85.40	
	Column %	17.87	24.23	
	Total %	3.36	19.67	
5	Count	323.00	5037.00	
	Row %	6.03	93.97	
	Column %	3.32	12.02	
	Total %	0.63	9.76	

2) Narcotic addiction

On the scree plot, the slope of the curve sharply increased at five clusters. Thus, the number of clusters was determined to be five (Fig. 25).

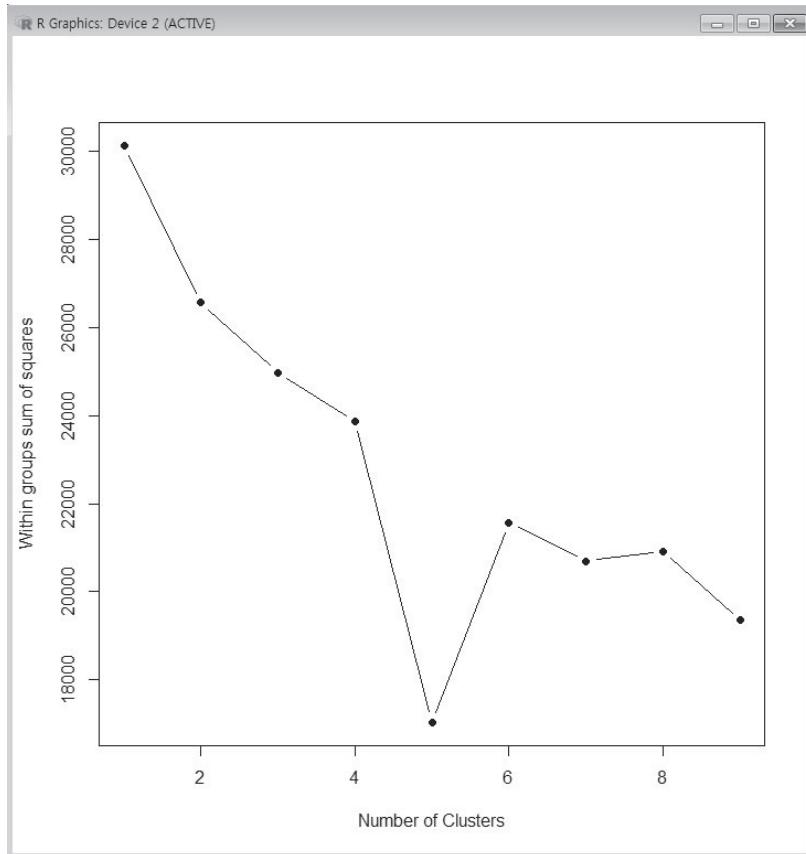


Fig. 25. Scree plot of cluster analysis (Narcotic addiction)

Factors with a cluster mean of 0.3 or higher were included in each cluster. Cluster 1 had 250,697 cases, but no factor was included, based on the aforementioned criterion. Cluster 2 had 3,129 cases and was labeled as "New drug." Cluster 3 had 1,099 cases and was labeled as "Hallucinogens." Cluster 4 had 1,332 cases and was labeled as "Cocaine." Finally, cluster 5 had 1,756 cases and was labeled as "Opium, Morphine" (Table 14).

Table 14 Cluster analysis (Narcotic addiction)

```

R Console

> fit = kmeans(clust_data, 5) # 5 cluster solution
> fit
K-means clustering with 5 clusters of sizes 250697, 3129, 1099, 1332, 1756

Cluster means:
          Opium   Morphine    Heroin   Cocaine   Codeine Amphetamine
1 0.00000000 0.00000000 0.001559652 0.000000000 0.0002153995 0.002648616
2 0.02684564 0.03004155 0.026526047 0.021732183 0.0169383190 0.034515820
3 0.02456779 0.01364877 0.020018198 0.030027298 0.0027297543 0.013648772
4 0.10810811 0.11336336 0.289039039 0.986486486 0.0195195195 0.289789790
5 0.55694761 0.51309795 0.068337130 0.007972665 0.0956719818 0.017653759

Benzodiazepines Lysergic_acid_diethylamide Cannabis Marihuana Propofol
1 0.03215435 0.0003510214 0.004635077 0.002042306 0.002572827
2 0.06839246 0.0092681368 0.030361138 0.010546500 0.123042506
3 0.02729754 0.0809827116 0.050955414 0.044585987 0.005459509
4 0.04429429 0.1231231231 0.125375375 0.177177177 0.015765766
5 0.03644647 0.0045558087 0.016514806 0.014806378 0.005125285

Precursor_chemical Ecstasy Stimulant Psychotropic_Drugs Hallucinogenics
1 0.007451226 0.0006142874 0.008129335 0.005619496 0.000000000
2 0.038350911 0.0281240013 0.023968319 0.232023011 0.009907319
3 0.030027298 0.0282074613 0.029117379 0.018198362 1.000000000
4 0.057807808 0.0645645646 0.224474474 0.108108108 0.089339339
5 0.023348519 0.0051252847 0.031890661 0.022209567 0.003416856

New_drug:
1 0.00000000
2 1.00000000
3 0.03366697
4 0.09234234
5 0.01537585

Clustering vector:

```

A detailed analysis was performed to identify the clusters affecting sentiment toward narcotics (negative and positive). The results showed that negative sentiment toward narcotics was the highest in cluster 2 (New drug: 49.30%), followed by cluster 4 (Cocaine: 20.08%)(Table 15).

Table 15 Segmentation analysis of cluster (Narcotic addiction)

		Attitude	
		0	1
fit.cluster			
2	Count	1662.00	1467.00
	Row %	53.12	46.88
	Column %	49.30	37.19
	Total %	22.72	20.05
3	Count	509.00	590.00
	Row %	46.31	53.69
	Column %	15.10	14.96
	Total %	6.96	8.06
4	Count	677.00	655.00
	Row %	50.83	49.17
	Column %	20.08	16.60
	Total %	9.25	8.95
5	Count	523.00	1233.00
	Row %	29.78	70.22
	Column %	15.51	31.25
	Total %	7.15	16.85

Visualization

The daily risk of respiratory symptoms is shown in Fig. 26.

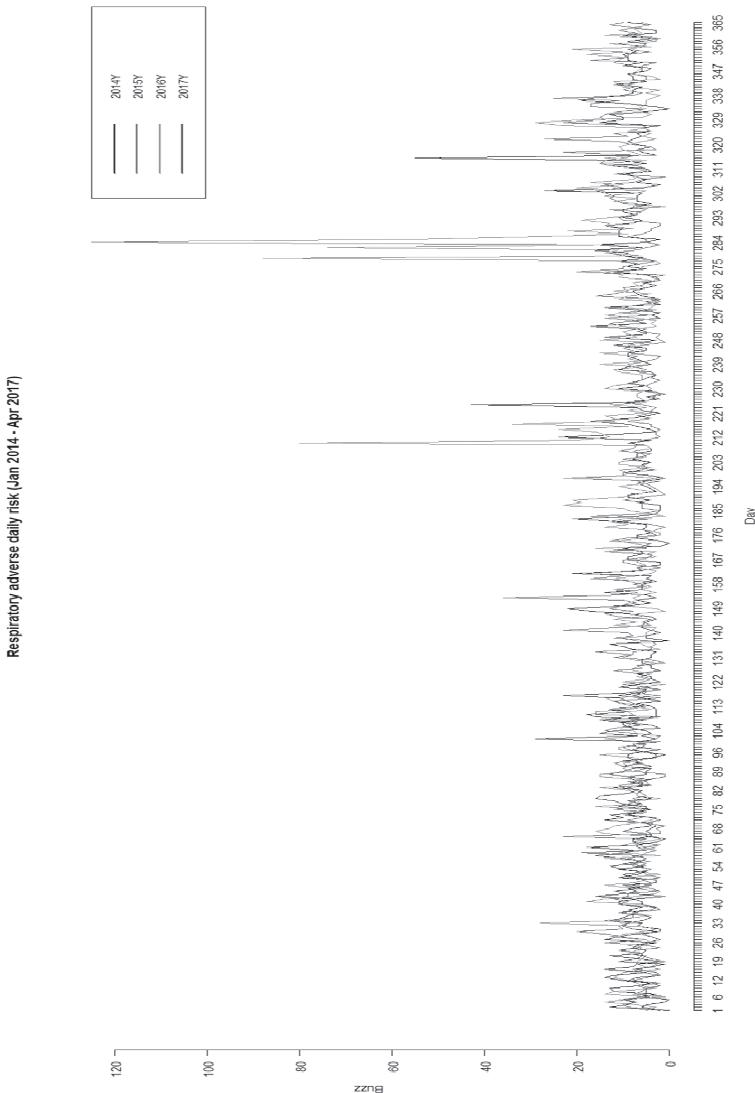


Fig. 26. Daily risk of respiratory symptoms

Discussion and Conclusion

Discussion

The objectives of the present study were to collect drug-related online documents from all accessible channels in South Korea, analyze the documents contained in social media data using machine learning approaches, and develop predictive models for drugs side effects and the risk of narcotic addiction. The results are summarized as follows.

First, regarding sentiment toward drugs (medications), 64.2% were positive, 7.1% neutral, and 28.7% negative. The most common respiratory side effect was coughing (25.1%), followed by asthma (20.5%) and sputum (12.8%). Regarding narcotics, the most frequent was cannabis (18.8%), followed by hallucinogens (17.9%), benzodiazepines (9.9%), new narcotic analogs (8.5%), and stimulants (7.5%).

Second, in searching for future signals for narcotics, hallucinogens, propofol, psychotropic drugs, cocaine, benzodiazepines, and stimulants were associated with strong signals, while codeine, ecstasy, and LSD were associated with weak signals. Signals that were strong but had a low increase rate were observed in cannabis and new narcotic analogs, and latent signals were found in morphine, heroin, opium, marijuana, precursor chemicals, and amphetamines. Particularly, propofol, with a strong signal in the first quadrant, also showed a high increase rate. Therefore, a system to manage propofol should be urgently put in place.

Third, the ROC curve to assess the machine learning models for respiratory symptoms showed that the random forest model, neural network model, and decision tree model were best, in that order. The ROC curve to evaluate the machine learning models for narcotics revealed that the models were not very good.

Fourth, regarding the association rules among respiratory symptoms, the association of the four variables {Cough, Sneeze, Sputum} => {Snot} was the strongest. Regarding the association rules among narcotic-related keywords, the association between the two variables {Heroin} => {Cocaine} was the strongest.

Conclusion

Based on the study findings, the following policy suggestions are made.

First, drug ontology should continuously be revised. The ontology used in the present study was a classification dictionary developed based on the data obtained from online channels for three years, starting from 2014. A

classification system was used for drug side effects. Hence, the ontology should be continuously modified and revised by adding keywords for new drugs and side effects on a yearly basis.

Second, the machine learning models developed in the present study should be made more intelligent through continuous learning. Namely, to advance the models, new data should be added and additional training be provided.

Third, data for use in the machine learning models should continuously be updated. When a machine learning model developed on training data is applied to test data, the predicted classification is different from the actual. Hence, to increase the model's prediction accuracy, high-quality training data should be generated by selecting only those cases whose actual and predicted classifications are identical, and the model should be re-trained on those data.

Fourth, the overall evaluation of the predictive models for narcotics was very low. While sentiment analysis on a topic with negative connotations, e.g., narcotics, should be based on a sentiment dictionary developed for the domain, the present study only measured positive and negative terms, without differentiating the emotions in further detail.

Fifth, similar to the theoretical drug-surveillance background utilizing social media, when general consumers discuss drugs and narcotics, they do not use professional terms, e.g., the classification system utilized in the present study. Therefore, a dictionary of colloquial terms and slang for medications and narcotics should be developed in the future.

Finally, if the present study findings are actually applied in the real world, an intelligent information service could be provided to detect the risks of drug-related incidents in advance and prevent recurrences.

References

- Aagaard L, Nielsen LH, Hansen EH. Consumer reporting of adverse drug reactions: a retrospective analysis of the Danish adverse drug reaction database from 2004 to 2006. *Drug Saf.* 2009;32:1067-1074.
- Ansoff, H. I. (1975). Managing strategic surprise by response to weak signals. *California Management Review*, 18(2), 21-33.
- Avery AJ, Anderson C, Bond CM, et al. Evaluation of patient reporting of adverse drug reactions to the UK Yellow Card Scheme: literature review, descriptive and qualitative analyses, and questionnaire surveys. Southampton: NIHR HTA; 2011. doi:10.3310/hta15200.
- Blenkinsopp A, Wilkie P, Wang M, Routledge PA. Patient reporting of suspected adverse drug reactions: a review of published literature and

- international experience. *Br J Clin Pharmacol.* 2007; 63(2):148-156. doi:10.1111/j.1365-2125.2006.02746.x
- Bouvy JC, De Bruin ML, Koopmanschap MA. Epidemiology of adverse drug reactions in Europe: a review of recent observational studies. *Drug Saf* 2015;38:43753. doi:10.1007/s40264-015-0281-0.
- Chung SY, Jung SY, Shin JY, Park BJ. The role of the KIDS for enhancing drug safety and risk management in Korea. *J Korean Med Assoc* 2012;55:861-868.
- Cossey M. Applied pharmacology. In Courtenay M, Griffiths M (Eds) Independent and Supplementary Prescribing: An Essential Guide. Second edition. Cambridge University Press, Cambridge;2010: 65-84.
- DailyStrength: Online Suppoprt Groups and Forums.
<https://www.dailystrength.org/> accessed May 18, 2017
- Ellene C, Baumgarten A. WHO Drug dictionaries and ATC, WHO-ART and MedDRA; Presentation for UMC PV course in Mysore. September 3, 2015.
<https://www.dropbox.com/sh/ombjtus3ovo22j5/AACftHSIaDN6btWSHfEPINsa?dl=0&preview=Terminologies,for,coding,-,Carin,%26,AnnaB.pdf>. Accessed May 25, 2017.
- Ernst FR, Grizzle AJ. et al. Drug-related morbidity and mortality: updating the cost-of-illness model. *J Am Pharm Assoc* 2001;41(2):192-9.
- Greener M. Understanding adverse drug reactions: an overview. *Nurse Prescribing*.2014; 12(4): 189-195.
- Hazell L, Shakir SA. Under-reporting of adverse drug reactions. *Drug Saf.* 2006;29:385396.
- Hiltunen, E. (2008). "The future sign and its three dimensions". *Futures* 40, 247–260.
- Hughes S, Cohen D. Can online consumers contribute to drug knowledge? A mixed-methods comparison of consumer-generated and professionally controlled psychotropic medication information on the internet. *J Med Internet Res.* 2011;13(3):e53.doi: 10.2196/jmir.1716.
- Jung, GH (2010). Future prediction method using text mining and network analysis. Korea Institute of S&T Evaluation and Planning.
- Koo HK(2008). "Effects of adverse drug reactions detected using monitoring program on the length of stay and charges in the hospital setting.,Doctor of Philosophy in Medicine. Seoul National University.
- Korea Institute of Drug Safety & Risk Management(2014). Reporting terms or drug side effects.
- Lardon J1, Abdellaoui R, Bellet F, et al. Adverse drug reaction identification and extraction in social media: a scoping review. *J Med Internet Res.* 2015;17(7):e171. doi: 10.2196/jmir.4304.

- Lazarou J, Pomeranz BH, Corey PN. Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *JAMA* 1998;279(15):1200-5.
- MedHelp-Health Community, health information, medical questions.
<http://www.medhelp.org/> accessed May 18, 2017
- Park, CG., & Kim, HJ (2015). A study on the development of energy new industry through internet- exploring future signals using text mining-, The Korea Energy Economics Institute.
- Park RW (2016). Establishment of safe use monitoring of pharmaceuticals using health insurance big data, National Health Insurance Service.
- PatientsLikeMe: Live better, together. <https://www.patientslikeme.com/> accessed May 18, 2017
- Pirmohamed M, James S, Meakin S et al. Adverse drug reactions as cause of admission to hospital: prospective analysis of 18,820 patients. *BMJ* 2004;329(7456):15-9.
- Sarker A, Ginn R, Nikfarjam A, et al. Utilizing social media data for pharmacovigilance: a review. *J Biomed Inform.* 2015;54(4):202-12. doi: 10.1016/j.jbi.2015.02.004.
- Sloane R, Osanlou O, Lewis D, et al. Social media and pharmacovigilance: a review of the opportunities and challenges. *Br J Clin Pharmacol.* 2015; 80(4):910-20. doi: 10.1111/bcp.12717.
- Spärck Jones, K. (1972). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". *Journal of Documentation*, 28, 11–21. doi:10.1108/eb026526
- Sultana J, Cutroneo P, Trifir G. Clinical and economic burden of adverse drug reactions. *J Pharmacol Pharmacother.* 2013;4:S73S77.
- Supreme Prosecutors' Office of the Republic of Korea(2015), 2015 Drug Criminal White Paper.
- Twitter. Welcome to Twitter. <https://www.twitter.com/> accessed May 18, 2017
- Uppsala Monitoring Centre. What is WHO-ART?. Updated Jan 9, 2017. <https://www.who-umc.org/vigibase/services/learn-more-about-who-art/>. Accessed May 25, 2017.
- Van Geffen ECG, van der Wal SW, van Hulten R, et al. Evaluation of patients' experiences with antidepressants reported by means of a medicine reporting system. *Eur J Clin Pharmacol.* 2007;63:11931199.
- Vilhelmsson A, Svensson T, Meeuwisse A, et al. What can we learn from consumer reports on psychiatric adverse drug reactions with antidepressant medication? Experiences from reports to a consumer association. *BMC Clin Pharmacol.* 2011;11:16.
- Yoo, S. H., Park, H. W., & Kim, K. H. (2009). A study on exploring weak

- signals of technology innovation using informetrics. *Journal of Technology Innovation*, 17(2), 109-130.
- Yoon, J. (2012). "Detecting weak signals for long-term business opportunities using text mining of Web news." *Journal Expert Systems with Applications*, 39(16), 12543-12550.
- [http://www.sciencetimes.co.kr/?p=110174&post_type=news&news-tag=
http://www.drugfree.or.kr/information/](http://www.sciencetimes.co.kr/?p=110174&post_type=news&news-tag=http://www.drugfree.or.kr/information/), Korean Association Against Drug Abuse. Accessed 2017. 7. 11.
- <http://www.biopharminternational.com/report-south-korea-0>

INDEX

- abline(), 79
- acceptance region, 43
- accidental sampling, 41
- accuracy, 186
- activation function, 151
- alpha(), 107
- alternative (research) hypothesis, 42
- anova(), 88
- apriori(), 171, 172
- array(), 15
- artificial intelligence, 118
- artificial neural networks, 149
- as.formula(), 127
- as.matrix(), 171
- association analysis, 170
- association measures, 57
- attach(), 21
- AUC (Area Under the Curve), 187
- barplot(), 248
- bartlett.test(), 66, 98, 99
- between-subjects Effect Test, 112
- biological (Human) neural network, 149
- bootstrap aggregating, 134
- boxplot(), 51, 55
- c(), 13
- Categorical data, 38
- cbind(), 26
- central tendency, 45
- chisq.test(), 58
- classifier, 134
- cluster analysis, 179
- cluster sampling, 41
- coefficient of determination, 87
- Coefficient of variance, 47
- colnames(), 126
- combination function, 151
- Conceptual definition, 36
- conditional statements, 19
- confidence, 40, 170
- confirmatory factor analysis, 97
- Constant, 38
- contingency coefficient, 57
- Continuous data, 38
- convenience sampling, 41
- cor.test(), 81
- correlation analysis, 80
- corrplot, 83
- Covariate variable, 38
- Cramer's V, 57
- CRAN mirrors, 7
- CRAN site (www.r-project.org), 7
- Cronbach's alpha, 103
- cross-tabulation analysis, 56
- ctab(), 53
- ctree(), 141
- cv.test, 60
- data mining, 118
- data.frame(), 22
- decision tree model, 141
- Decision Tree Model Evaluation, 205
- Decision Trees ROC, 212
- Deductive, 35
- deep learning, 118
- Dependent variable, 38
- descriptive statistics, 44
- dimnames(), 127
- dispersion, 46
- do.call(), 173
- downward accuracy, 187
- ecological fallacy, 39
- eigenvalue, 98
- ensemble, 134
- Enter method, 89

- equal-variance assumption, 62, 65
- equivariance, 89
- error rate, 186
- exploratory factor analysis, 97
-
- factanal(), 99
- factor analysis, 97
- factor loadings, 98
- factor rotation, 98
- factor scores, 103
- false negative, 186
- false positives, 186
- Filter(), 173
- FPR (False Positive Rate), 187
- frequency Analysis, 53
- ftable(), 53
- F-test, 65
- function(), 17
-
- geographical data, 250
- glht(), 67
- glm(), 131
- graph.edgelist(), 173
-
- hidden layers, 150
- hist(), 51
- homogeneity test, 57
- hypothesis testing, 42
-
- independence test, 56
- Independent variable, 38
- independent-sample T-test, 62
- individualistic fallacy, 39
- Inductive, 36
- inferential statistics, 44, 56
- input layer, 150
- inspect(), 172
- install.packages("arules"), 171
- install.packages('foreign'), 24
- install.packages('Rcmdr'), 47
- install.packages("arulesViz"), 175
- install.packages("igraph"), 173
- install.packages('caret'), 165
- install.packages('catspec'), 53
- install.packages('dplyr'), 31, 173
- install.packages('e1071'), 126
-
- install.packages('gmodels'), 59
- install.packages('gplots'), 69
- install.packages('kernlab'), 165
- install.packages('lm.beta'), 86
- install.packages('mapproj'), 253
- install.packages('maps'), 251
- install.packages('multcomp'), 66
- install.packages('neuralnet'), 155
- install.packages('NeuralNetTools'), 154
- install.packages('nnet'), 115, 132, 153
- install.packages('party'), 141
- install.packages('partykit'), 143
- install.packages('ppcor'), 84
- install.packages('ROCR'), 158
- install.packages('wordcloud'), 236
- interaction effect, 75
- interaction plot, 77
- Interval scale, 37
-
- judgment sampling, 41
-
- Kendall's τ , 57
- kmeans(), 182
- Kurtosis, 47
-
- labels(), 173
- laplace smoothing, 125
- length(), 53
- library(MASS), 26
- library(RColorBrewer), 236
- lift, 170
- lines(), 51
- List, 16
- lm(), 86
- lm.beta(), 86
- logistic regression analysis, 112
- logistic regression model, 130
- Logistic Regression Model Evaluation, 196
- Logistic ROC, 210
- logit model, 130
- loop statements, 20

- machine learning, 118
- machine learning model evaluation, 186
- machine learning training data, 122
- manova(), 109
- matrix(), 15
- Mean, 45
- Mean Decrease Accuracy (%IncMSE), 134
- Mean Decrease Gini (IncNodePurity), 134
- Mean deviation, 46
- measurement rule, 37
- Median, 46
- Mediating effect, 38
- Mediating variable, 38
- memory.size(), 140
- merge(), 28
- Mode, 46
- model function, 134
- Moderation variable, 38
- Multicollinearity, 89
- Multilayer neural networks, 150
- multiple comparisons, 65
- multiple regression analysis, 89

- Naïve Bayes classification model, 124
- Naïve Bayes Classification Model Evaluation, 189
- NaïveBayes ROC, 208
- naiveBayes(), 127
- Natural science, 35
- Neural Network Model Evaluation, 193
- Neural Networks ROC, 210
- neuralnet(), 156
- Nominal scale, 37
- non-probability sampling, 41
- null hypothesis, 42
- numSummary(), 49

- odds ratio, 130
- OLS (Ordinary Least Squares), 89
- one-sample T-test, 61
- one-way ANOVA, 65
- open-source, 1

- Operational definition, 36
- Ordinal scale, 37
- output layer, 150
- overfitting, 152

- paired T-test, 64
- parallel coordinates plot, 175
- parameter, 40
- partial correlation analysis, 84
- partial regression residual plots, 89
- paste(), 127
- pcor.test(), 84
- Pearson's R, 57
- Percentiles, 46
- performance(), 158
- plot(), 239
- plot.igraph(), 173
- plotmeans(), 69
- plotnet(), 154
- polychotomous logistic regression, 115
- population, 40
- posterior probability, 124
- post-hoc analysis, 65
- predict(), 127
- prediction(), 158
- prior probability, 124
- Probability sampling, 40
- purposive sampling, 41
- p*-value, 43

- qualitative, 37
- quality(), 173
- quantitative, 37
- Quartiles, 46
- quota sampling, 41

- R², 87
- random forest model, 134
- Random Forest Model Evaluation, 202
- Random Forests ROC, 211
- randomForest(), 135
- Range, 46
- Ratio scale, 37
- rbind(), 28

- read.spss(), 24
- read.table(), 23
- reductionism fallacy, 39
- reinforcement learning, 120
- rejection region, 43
- representative value, 45
- result[[1]], 156
- ROC curves, 187
- round(), 196
- sample, 40
- sample size, 17
- sample(), 41
- sampling error, 40, 42
- sapply(), 147
- Scale, 37
- scatter diagram, 78
- Scheffé, 65
- scree chart, 98
- segmentation analysis, 184
- sensitivity, 186
- seq(), 14
- sigmoid function, 151
- significance, 43
- simple random sampling, 40
- simple regression analysis, 86
- single-layer neural network
 - (perceptron), 150
- Skewness, 47
- snowball sampling, 41
- Social science, 35
- Somers' D, 57
- Specificity, 186
- Standard deviation, 47
- standard score, 18
- standardized regression coefficients, 86
- statistics, 40
- step(), 89
- Stepwise method, 89
- stratified random-cluster sampling, 41
- stratified sampling, 41
- summary(), 66
- summary.aov(), 112
- supervised learning, 119
- support, 170
- support vector machine (SVM), 162
- Support Vector Machine Model
 - Evaluation, 199
- SVM ROC, 211
- svm(), 164
- systematic sampling, 41
- t.test(), 62
- tapply(), 65
- text(), 240
- tolerance, 94
- TPR (True Positive Rate), 187
- true negative rate, 186
- true positive rate, 186
- Tukey, 65
- two-way ANOVA, 75
- type-I error, 43
- type-II error, 43
- unit of analysis, 39
- unsupervised learning, 119
- upward accuracy, 187
- Validity, 97
- Value, 38
- Variables, 38
- Variance, 47
- varImpPlot(), 138
- Vector, 13
- VIF (Variance Inflation Factor), 89
- Visualization, 236
- wordcloud(), 237
- write.matrix(), 26
- η (Eta), 57
- χ^2 -test, 56