```
####### How to install R using Homebrew:(맥북 운영체제만 해당)
# $brew install r or $brew cask install rstudio
# $sudo r
####### How to install R packages:
# $sudo r
# $install.packages("gplot2")
####### How to run R script:
# $Rscript your-file.R
# Load packages
library('ggplot2') # visualization
library('ggthemes') # visualization
library('scales') # visualization
library('dplyr') # data manipulation
library('randomForest') # classification algorithm
# Reading the data set (lottery-train.csv)
train <- read.csv('./lottery-train-2023.csv', stringsAsFactors = F)
# STEP 1: Split the data set into two parts: 'train' and 'test'
# your train data should be more 70% of the given data set
# show your code below
# e.g.) full[1:1500,]
# Display your datasets. Do Not Delete.
str(train)
str(test)
# Build a random forest mode
# STEP 2-1: Set a random seed
# show your code below
# STEP 2-2: Build your random forest model
# You are free to choose your selection of available feature.
# See more detailed explanation in the midterm instruction.
# Identify your target feature to predict and the rest.
# BONUS: If you have your own feature to add, make one and use it in your code.
#          Explain your feature here (add comments).
# show your code below
# e.g.) randomForest(factor(?) ~ ? + ?, data = ?)
# STEP 3: Show model error, calculate importance, and rank valuable. Draw figures.
# STEP 3-1: Show model error and plot it.
# Your output figures should be close to the examples in the midterm instruction.
# show your code below
# STEP 3-2: Calcuate importance
# show your code below
# STEP 3-3: Create a rank variable based on importance. And use ggplot to draw figures as in the midterm instruction.
# show your code below
# Prediction using your_model against test data. Do Not Delete.
prediction <- predict(your_model, test)
# STEP 4: Caculate solution (using prediction above) and save it to 'lottery-solution.csv' where it contain the two columns: round and win.
# show your code below
```