# STA 141A Final Project

Luc Chen, Ali Taleghani, Yirong Xu, Li Yuan, and Jianing Zhu

6/6/2022

## Contents



Group Member Contribution:

| Name | Contribution |
|---|---|
| Luc Chen | Part III |
| Ali Taleghani | Part IV |
| Yirong Xu | Part V |
| Li Yuan | Part I & II |
| Jianing Zhu | Part VI |

# I. Introduction and Research Questions

Sometimes website builders and administrators may want to know how long people spend on their websites. It is very important to know this information because it can help website builders and administrators improve their websites and discover more potential business value.

There are a lot of aspects that can affect the time people spend on a website. For example, how people visit the website. They may directly type the domain name in their browser and visit, or they can click on the link from search engine search results. It may affect the time people spend on the website because people might be more interested in the website if they get it from the search engine. Moreover, based on a lot of marketing research, the loading time seriously affects user stickiness. The less loading time, the more time people spend on the website. There are some other factors such as number of hits, size of the website, and usage of Content Delivery Network (CDN), which can improve the connection speed and save bandwidth.

There is a website called Sample Academy, which mainly focuses on academics, providing well-made learning materials for students and help students learn easier. We want to know whether the reference, average loading time, number of hits, how much data transferred from server to visitors, and usage of CDN affect the Sample Academy website's user stickiness from November 23, 2020 to May 1, 2022. Moreover, we want to test whether the bot detection algorithm works well behind the Sample Academy website. It is very important because a good bot detection algorithm can defend against cyber attacks.

To figure out the relationship between the time people spend on the website (response variable) and other factors such as the reference, average loading time, number of hits, size of the website, and usage of CDN (independent variables), we will use multiple linear regression. If the data contains some potential problems that make the assumptions fail, we will use remedies to fix them such as removing influential points and using transformations. We will use logistic regression to figure out the accuracy of the bot detection algorithm behind the website. We will mainly use the number of hits and the average loading time as independent variables because these are the most significant factors in determining bot detection strength. Furthermore, the bots usually hit very frequently on websites and have high bandwidth so they hit more times than humans (hits) and have a smaller average loading time (loadtime).

# II. Data Information

Before figuring out what factors really affect the time people spend on the Sample Academy website and how well the bot detection algorithm works, we need to know some basic aspects of the data we have.

- `duration`: The total time, in second(s), each visitor spends on the Sample Academy website on that day.
- `hits`: The total number of hits on the Sample Academy website of each visitor on that day.
- `loadtime`: The average loading time, in millisecond(s), of all the hits done by each visitor on the Sample Academy website on that day.
- `datatransferred`: The total data transferred, in megabyte(s), from the Sample Academy website to each visitor on that day.
- `cdn`: The Content Delivery Network provider that each visitor randomly connects to. There are four CDN providers: `Cloudflare` (6,016 counts), `Beluga` (6,044 counts), `DDOS-GUARD`(5,712 counts), and `Microsoft` (5,976 counts).
- `reference`: The method each visitor accessed the Sample Academy website. There are two methods: `Direct` (12,054 counts), visit by typing domain name in the browser and `External` (11,794 counts), visit by clicking on the link from search engines search results.

- `bot`: The suspected type of visit of each visitor. There are two categories: `Bot` (11,238 counts) means that the visit looks like a bot and `Human` (12,6107 counts) means that the visit looks like a real human.

Figure 1: Summary Statistics of All Numeric Variables

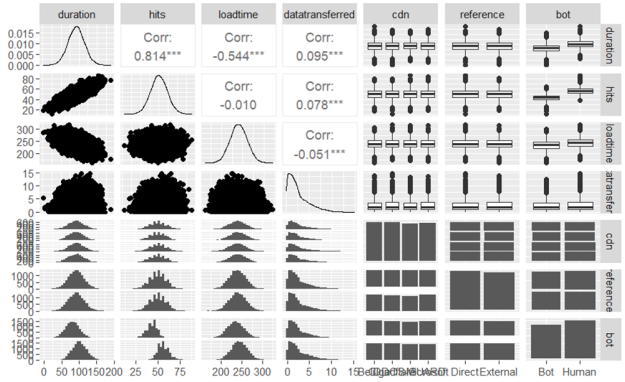
| | hits | loadtime | datatransferred | duration |
|---|---|---|---|---|
| Minimum | 12.00 | 161.93 | 0.01 | 0.20 |
| Mean | 50.08 | 240.08 | 2.55 | 92.03 |
| Maximum | 86.00 | 320.28 | 14.81 | 187.84 |
| Standard Deviation | 9.02 | 19.94 | 2.23 | 22.32 |
| Length | 23848.00 | 23848.00 | 23848.00 | 23848.00 |

Each variable has 23,848 items. There may have outliers since the standard deviation of `duration` is big.

# III. Data Visualization

In order to do the multiple linear regression and the logistic regression, we first need to visualize our whole dataset by looking at Figure 2.

Figure 2: Scatterplot Matrix of Sample Academy Website Data




The x-axis represents the variables of the columns, and the y-axis represents the variables of the rows.

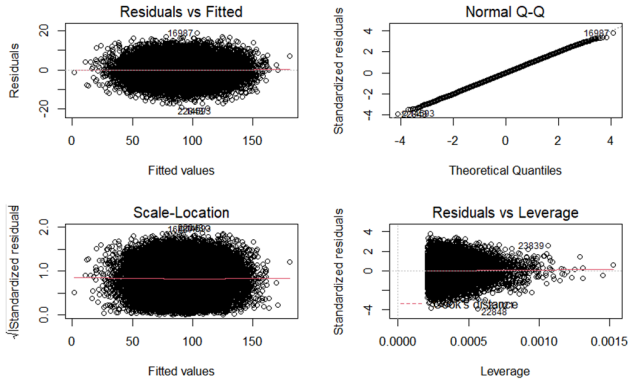
Among those numerical variables, we can see `hits` and `loadtime` are related to `duration`. Due to the distributed denial-of-service (DDoS) protection provided by those CDNs, most recorded visitors here look fine and most of our data look normal except `datatransferred`. Seems in most cases, there are only a few data transferred from the server to visitors.

Moreover, from the `bot` column, we can see human-like visits generally have higher means in `duration`, `hits`, and `datatransferred`, which makes sense. Real people may spend more time than robots on the website because people need time to read the website content and this causes increases in `duration`, `hits`, and `datatransferred`.

## 1. Linear Regression Model Assumption Diagnosis

We first need to check whether the assumptions for multiple linear regression hold. Figure 3 is the diagnostic plots of our model: $\hat{duration} = \hat{\beta}_0 + \hat{\beta}_1 hits + \hat{\beta}_2 loadtime + \hat{\beta}_3 datatransferred + \hat{\beta}_4 reference + \hat{\beta}_5 cdn$.

Figure 3: Diagnostic Plots of the Predicted Linear Regression Model




- **Linearity**: According to the Residuals vs Fitted plot in Figure 3: Diagnostic Plots of the Predicted Linear Regression Model, the residuals are randomly distributed around the red line, where residual equals to zero. This suggests that the linearity assumption holds.

- **Normality**: In order to figure out whether the normality assumption holds, we use Kolmogorov-Smirnov test, where we assume that the dataset is normally distributed. By performing the test, we get p-value equals to 0.8444381, which is an extremely large p-value, telling us that our dataset may be normally distributed. Therefore, our assumption of normality holds.

- **Independence**: In order to figure out whether the observations are independent, we use Durbin-Watson test, where we assume that each observation is independent in the dataset. After performing the test, we get p-value equals to 0.9943169, which is a very large p-value, meaning that in reality our observations are uncorrelated, independent. Thus, our independence assumption holds.

- **Homoscedasticity**: To check whether our model has constant variance, we use studentized Breusch-Pagan test, where we assume our residuals have constant variance. By doing the test, we get p-value equals to 0.6310332, which is extremely large, telling that our residuals have constant variance. As a result, our homoscedasticity assumption holds.

- **Multicollinearity**: We calculate the variance inflation factor (VIF) of each variable by using the formula $VIF_i = \frac{1}{1-R_i^2}$ ($R_i^2 = \frac{SSR}{SSTO}$, which is a goodness-of-fit measure for linear regression models, higher $R_i^2$ means better model) to see whether they are correlated with each other. After computing the VIFs, we find out all of them are around 1, smaller than any threshold we usually use (5 as an example). As a result, our explanatory variables do not have multicollinearity.

- **Outlier**: We use regression deletion diagnostics function and the Residuals vs Leverage plot in Figure 3 to check whether we have outliers in our dataset, and we find 51 outliers. We can spot the outliers by looking at the Residuals vs Leverage plot in Figure 3. Obviously, points 22848, 22088, and so on are outliers. Since these outliers may affect our model a lot, we remove them to make our prediction more accurate.

- **High Leverage**: According to the Residuals vs Leverage plot in Figure 3, we cannot find any point that is far away from the big cluster, so there are no obvious high leverage points.

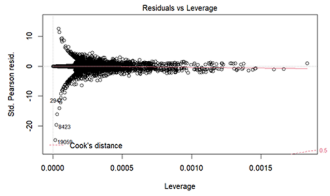
In summary, all of our assumptions for the multiple linear regression hold. In order to make our model more accurate, we remove these 51 outliers, since outliers usually make our fitted model contain more error and make our prediction less credible.

## 2. Logistic Regression Model Assumption Diagnosis

Before doing the logistic regression, We need to check whether the assumptions for logistic regression hold. Our model is: $P = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 hits+\hat{\beta}_2 loadtime}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 hits+\hat{\beta}_2 loadtime}}$. In order to make sure our logistic regression model is accurate enough, our test set has 5,962 observations, and our train set has 17,886 observations.

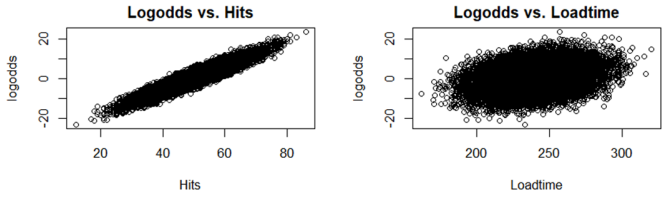
- **Independence**: In order to figure out whether the observations are independent, we use Durbin-Watson test, where we assume that each observation is independent in the dataset. After performing the test, we get p-value equals to 0.543645, which is a very large p-value, meaning that in reality our observations are uncorrelated, independent. Thus, our independence assumption holds.
- **Binary:** The response variable, `bot`, is binary taking values either `human` or `bot`. The assumption that the dependent variable is binary holds.
- **Multicollinearity**: We calculate the variance inflation factor (VIF) of each variable by using the formula $VIF_i = \frac{1}{1-R_i^2}$ ($R_i^2 = \frac{SSR}{SSTO}$, which is a goodness-of-fit measure for linear regression models, higher $R_i^2$ means better model) to see whether they are correlated. After computing the VIF's, we find out both of them are 1.914857, smaller than the threshold we usually use (5 as an example). As a result, our explanatory variables `hits` and `loadtime` do not have mlticollinearity.
- **Sample Size:** The assumption for large sample size holds because we have 2,3848 observations in total.
- **Outlier**: We cannot find any extreme outliers or influential points from Figure 4: Residuals vs Leverage for Logistic Regression Model. As a result, our model does not have extreme outliers.

Figure 4: Residuals vs Leverage for Logistic Regression Model




- **Linearity**: If we look at Figure 5: Scatterplot of Hits and Loadtime vs Logodds of Bot, we can see that there are linear relationships between explanatory variables (`hits`, `loadtime`) and the logit of the response variable (`bot`). Therefore, the linearity assumption for logistic regression holds.

Figure 5: Scatterplot of Hits and Loadtime vs Logodds of Bot




In summary, all assumptions for the logistic regression hold. We can start perform the analysis now.

# IV. Analysis

## 1. Multiple Linear Regression to Predict Time People Spend

Before performing the multiple linear regression to predict how much time people spend on the Sample Academy website, we hypothesize that the number of hits, average loading time, how much data transferred from the server to visitors, website reference sources, and CDN do not affect the time people spend on the website ($H_0 : \hat{\beta}_{hits} = \hat{\beta}_{loadtime} = \hat{\beta}_{datatransferred} = \hat{\beta}_{reference} = \hat{\beta}_{CDN} = 0$). In contrast, at least one of these factors has relationship with the time people spend on the website ($H_1$ : At least one $\hat{\beta}_i \neq 0$, where $i$ represents each variable).

By performing F test ($F^* = \frac{MSR}{MSE}$, where MSE means mean square error and MSR means mean square due to regression) based on the whole multiple linear regression model, we get the test statistic $F^* = 66320$ with degrees of freedom 7 and 23789 and the p-value equals to $2.2 \times 10^{-16}$, which is too small, meaning that there is at least one of these factors has relationship with the time people spend on the website.

In order to know which variable(s) really affect(s) the time people spend on the website, we need to perform t test ($t^* = \frac{\hat{\beta}_i - \beta_i}{se(\hat{\beta}_i)}$, where $\hat{\beta}_i$ are the predicted coefficients, $\beta_i$ are the true or assumed coefficients, and $se(\hat{\beta}_i)$ are the standard errors of the predicted coefficients) on each independent variable based on the multiple linear regression model. We hypothesize that either the number of hits, average loading time, how much data transferred from the server to visitors, website reference sources, or CDN does not affect the time people spend on the website ($H_0 : \beta_i = 0$, where $i$ represents each variable). Otherwise, the independent variable does affect the time people spend on the website ($H_1 : \beta_i \neq 0$, where $i$ represents each variable).

By performing several t tests, we find out the following results in Figure 5: t test Results Based on the Multiple Linear Regression Model, where "Estimated Coefficient" means the predicted relationship with the response variable. If the estimated coefficient of an independent variable equals to zero, then it means there is no linear relationship between the independent variable and the response variable.

Figure 5: t test Results Based on the Multiple Linear Regression Model

| Indep. Variable | Estimated Coefficient | t statistic | p-value |
|---|---|---|---|
| Number of Hits | 2.000952416702 | 563.129576231227 | 0 |
| Loading Time | -0.599411758460554 | -373.84949990827 | 0 |
| Data Transferred | 0.046869333766552 | 3.25599245186985 | 0.00113154806132618 |
| External Reference | -0.0233828263182825 | -0.366049278803019 | 0.714331559374236 |
| Cloudflare CDN | 0.0131121371972422 | 0.146009923775936 | 0.883914791479309 |
| DDOS-GUARD CDN | 0.211625530343559 | 2.33525149815849 | 0.0195385555142657 |
| Microsoft CDN | 0.0401033707265305 | 0.445770486368566 | 0.655767099685131 |

As we can see from Figure 5 above, `hits`, `loadtime`, `datatransferred`, and `cdnDDOS-GUARD` have very small p-values, while other variables have relatively big p-values.

## 2. Logistic Regression to Check Bot Detection Algorithm

Before performing the logistic regression to determine whether the bot detection algorithm performs well behind the Sample Academy website, we hypothesize that either the number of hits or the average loading time has no effect on the bot detection algorithm ($H_0 : \beta_i = 0$, where $i$ represents each variable). Otherwise, at least one of these independent variables affects the bot detection algorithm ($H_1 : \beta_i \neq 0$, where $i$ represents each variable).

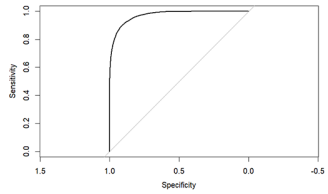
By doing two z tests ($z^* = \frac{\hat{\beta}_i - \beta_i}{se(\hat{\beta}_i)}$, where $\hat{\beta}_i$ are the predicted coefficients, $\beta_i$ are the true or assumed coefficients, and $se(\hat{\beta}_i)$ are the standard errors of the predicted coefficients) on the number of hits and the average loading time, we obtain test statistics $z^* = 57.71$ and $z^* = 43.50$ for the number of hits and average loading time respectively. Both estimators have p-values almost equal to 0.

Figure 6: Confusion Matrix of the Test Set

| True/Predicted | Bot | Human |
|---|---|---|
| Bot | 2,494 | 307 |
| Human | 296 | 2,865 |

In Figure 6, we create a confusion matrix of our test set, which represents the counts of all combination of values between the predicted label and the true label. By looking at the confusion matrix, we get error rate equals to $\frac{307+296}{2494+2865+307+296} = 0.1011406$, false positive rate equals to $\frac{307}{2494+307} = 0.1096037$, and false negative rate equals to $\frac{296}{2865+296} = 0.09364125$. The three results are all very small, which means that for most cases the model correctly predicts the classes, but there are still a few mislabeled data points.

Figure 7: ROC Curve of the Classification




We also perform the receiver operating characteristic curve (ROC) for testing the goodness of fit and the area under the curve (AUC) is 0.9724, which is very high.

# V. Result and Interpretation

## 1. Result of Multiple Linear Regression to Predict Time People Spend

From the F test in the multiple linear regression, we can see that the p-value equals to $2.2 \times 10^{-16}$, which is too small, meaning that there is at least one of these factors (`hits`, `loadtime`, `datatransferred`, `reference`, and `cdn`) has relationship with the time people spend on the website.

After performing the F test we figure out which variable(s) really affect(s) the time people spend on the website by performing several t tests based on the multiple linear regression model. From Figure 5: t test Results Based on the Multiple Linear Regression Model, we reject our hypotheses of variables `hits`, `loadtime`, `datatransferred`, and `cdnDDOS-GUARD`, since they have effect on the time people spend on the website. However, the other variables may not have effect on the time people spend on the website since their p-values are relatively large, larger than any significance levels we usually use (for example, 0.05).

To be specific, `hits` and `loadtime` are the most two significant variables because their p-values are almost equal to zero, meaning that the number of hits and loading time affect a lot on the time people spend on the website. If `hits` or `loadtime` does not affect the time people spend on the website, the probabilities we observe our test statistic $t^*$ or more extreme are almost equal to 0. Moreover `datatransferred` and `cdnDDOS-GUARD` are also significant because their p-values are 0.00113 and 0.01954, which are very small, smaller than any significance levels we usually use, meaning that how much data transferred from the

server to visitors and the DDOS-GUARD CDN service also affect the time people spend on the website. If `datatransferred`does not affect the time people spend on the website, the probability we observe our test statistic $t^*$ or more extreme is about 0.00113. In addition, if `cdnDDOS-GUARD`does not affect the time people spend on the website, the probability we observe our test statistic $t^*$ or more extreme is about 0.01954.

Despite knowing which variables affect the time people spend on the website, we can also see that if there is 1 more hit on the website, people may spend 2 more seconds in average on the website (other variables fixed), which makes sense because more hits means more times people access the website and that increases the time they stay. Moreover, if there is 1 millisecond increase in the average loading time, people may spend 0.599412 less second in average on the website (other variables fixed), which also makes sense because if the website is really slow, people may lose interest on it. In addition, if there is 1 more megabyte data transferred from the server to visitors, people may spend 0.046869 more second in average on the website (other variables fixed) since more data transferred means the visitor view more content, which indicates the visitor's interest on the website. Last but not least, the unit change of the time people spend on the website with DDOS-GUARD CDN service is higher than the time people spend on the website with Beluga CDN service 0.211626 in average.

### 2. Result of Logistic Regression to Check Bot Detection Algorithm

By performing two z tests on the number of hits and average loading time, we get both of the p-values almost equal to 0. As a result, the variables `hits` and `loadtime` are statistically significant, and have relationship with the bot detection algorithm. This means that if `hits` or `loadtime` does not affect the bot detection algorithm, the probabilities we observe our test statistic $z^*$ or more extreme are almost equal to 0. This indicates that our logistic regression model can predict the classification of bot and human very well.

From Figure 6: Confusion Matrix of the Test Set, we can see that the error rate is 0.1011406, meaning that in 10.11406% of all the visitors, some of them are human but classified as bot, some of them are bot but classified as human. The false positive rate is 0.1096037, meaning that there are 10.96037% of bot visitors are incorrectly classified as human. The false negative rate is 0.09364125, meaning that there are 9.364125% of human visitors are incorrectly classified as bot. Although there is still a possibility that the bot detection algorithm mislabeled data points, the majority of the true bot and human are well identified.

Moreover, from Figure 7: ROC Curve of the Classification, we can see the ROC curve is very smooth and hugs the top left corner very well, meaning that our classifier is good. By obtaining the value of AUC, which is 0.9724, indicating that most visitors are correctly classified as human and bot and our model and the bot detection algorithm perform very well.

## VI. Conclusion and Future Discussion

In this project, we used multiple linear regression and logistic regression techniques to figure out the factors that affect the duration of visitors on the website and to test the accuracy of bot detection algorithms. We first checked the assumptions and removed the outliers that highly affect the dataset, which improved the accuracy of linear regression coefficient estimates. Then multiple linear regression is been used to figure out the relationship between the total time each visitor spent on the Sample Academy website (response variable) and other factors such as the average loading time, total number of hits, method each visitor accessed, Content Delivery Network provider, and total data transferred (independent variables). Moreover, under the assumptions of logistic regression, we use the number of hits and the average loading time (independent variables) to predict visitor types (response variable), whether it is bot or human, and predict the accuracy of the bot detection algorithm behind the website.

We run the model and conclude that the total number of hits, average loading time, total data transferred, and DDOS-GUARD CDN provider are significantly key factors towards duration, meaning there are linear relationships between these variables and the duration time.

In addition to figuring out what affects the time people spend on the website, we also find the bot detection algorithm performs well behind the scenes of the Sample Academy website, since we obtained a low error rate, a low false positive rate, and a low false negative rate after constructing a confusion matrix (all rates are between 9% and 11%); and we got a high AUC value (0.9724) when graphing the ROC curve.

In the future, the website administrator can improve the Sample Academy website and discover even more potential commercial value by using DDOS-GUARD CDN more frequently, increasing the server's processing speed to decrease the average loading time, and adding more interesting content to increase data transferred from the server to visitors and number of hits. Moreover, the website administrator may still need to improve the bot detection algorithm, although it has very low error rate now, it still can be improved. Take Cloudflare as an example, its bot detection algorithm only has 0.01% of false positive rate, which can bring more security to website. The website administrator can consider to use those algorithms developed by tech companies such as Cloudflare, Microsoft, Yandex, and so on. It's crucial because a solid bot identification algorithm can protect the website from cyber-attacks.

## Appendix: R Script

```
knitr::opts_chunk$set(echo = TRUE)
rm(list=ls())
library(knitr)
library(ggplot2)
library(dplyr)
library(car)
library(lmtest)
library(MASS)
library(class)
library(GGally)
library(pROC)
################################################################################
#####I. Introduction and Research Questions#####################################
################################################################################
# Load the data
website = read.csv("https://tinyurl.com/sta141a-project-dataset",
                   header = TRUE)
################################################################################
#####II. Data Information#######################################################
################################################################################

# We created a function called numeric.summary.table here,
# which takes a data frame, to generate a nice table, which
# contains the five-number summary of all the numeric variables
# in the dataset.

# The function skips the categorical variables and always only
# considers the five-number summary of the numeric variables.

# The order of the variables does not matter so it works for
# almost all datasets.
numeric.summary.table = function(data){
  numeric.index = c()
  numeric.name = c()
  categorical.index = c()
  # Determine the type of the columns of data
  for (j in 1:length(data)){
    # If it is numeric, we will take it
    if (class(data[, j]) != "character"){
      numeric.index = append(numeric.index, j)
      numeric.name = append(numeric.name, colnames(data)[j])
    }
    # If it is not numeric, we won't use it, just store the index
    else{
      categorical.index = append(categorical.index, j)
    }
  }
  mins = c()
  means = c()
  maxs = c()
  sds = c()
```

```
  lens = c()
  numeric.result = NULL
  # Creating the five-number summary table
  for (i in 1:length(numeric.index)){
    mins[i] = round(min(data[,numeric.index[i]]),2)
    means[i] = round(mean(data[,numeric.index[i]]),2)
    maxs[i] = round(max(data[,numeric.index[i]]),2)
    sds[i] = round(sd(data[,numeric.index[i]]),2)
    lens[i] = length(data[,numeric.index[i]])
  }
  numeric.result = matrix(c(mins, means, maxs, sds, lens), byrow = TRUE,
                          ncol = length(numeric.index))
  colnames(numeric.result) = numeric.name
  rownames(numeric.result) = c("Minimum","Mean","Maximum","Standard Deviation","Length")
  kable(numeric.result)
}
numeric.summary.table(website)
##############################################################################
#####III. Data Visualization and Assumptions Check#############################
##############################################################################
ggpairs(website[, c(7, 1, 2, 3, 4, 5 ,6)])
linear.model = lm(duration ~ hits + loadtime + datatransferred + reference + cdn,
                  data = website)
# Diagnostic plots on first model
par(mfrow = c(2, 2))
plot(linear.model)
# Kolmogorov-Smirnov Test
ks.test = ks.test(linear.model$residuals, "pnorm",
        mean=mean(linear.model$residuals), sd=sd(linear.model$residuals))
ks.test
# Independence test
dwtest = dwtest(linear.model)
dwtest
# Constant variance test
bptest = bptest(linear.model)
bptest
# Multicollinearity
vif(linear.model)
# Find outliers
outlier.index = as.numeric(names(rstandard(linear.model)[rstandard(linear.model) < -3
                                          | rstandard(linear.model) > 3]))
outlier.index
set.seed(934)
# Create test and train sets
test.index = sample(seq_len(nrow(website)), size = nrow(website)/4)
test = website[test.index, ]
train = website[-test.index, ]
# Perform logistic regression
logistic.model = glm(as.factor(bot) ~ hits + loadtime,
                     data = train,
                     family = binomial)
dwtest = dwtest(logistic.model)
dwtest
```

```r
vif(logistic.model)
plot(logistic.model, which = 5)
p = predict(logistic.model, type = "response")
logodds =  log(p / (1-p))
par(mfrow = c(2, 2))
plot(logodds ~ train$hits, xlab = "Hits", main = "Logodds vs. Hits")
plot(logodds ~ train$loadtime, xlab = "Loadtime", main = "Logodds vs. Loadtime")
################################################################################
#####IV. Analysis################################################################
################################################################################
# Remove outliers and high leverage points and generate new data frame
new.website = website[-outlier.index, ]
# Fit new linear model
new.linear.model = lm(duration ~ hits + loadtime + datatransferred + reference + cdn,
                      data = new.website)
summary(new.linear.model)
t.test.result = matrix(c(c("Number of Hits", "Loading Time",
                           "Data Transferred", "External Reference",
                           "Cloudflare CDN", "DDOS-GUARD CDN",
                           "Microsoft CDN"),
                         summary(new.linear.model)$coefficients[-1,1],
                         summary(new.linear.model)$coefficients[-1,3],
                         summary(new.linear.model)$coefficients[-1,4]),
                       nrow = 7, ncol = 4)
colnames(t.test.result) = c("Indep. Variable", "Estimated Coefficient",
                           "t statistic", "p-value")
kable(t.test.result)
# Perform logistic regression
logistic.model
summary(logistic.model)
# Confusion matrix on test set
predicted = ifelse(predict(logistic.model, type = "response", test) > 0.5,"Human","Bot")
confusion = table(predicted,factor(test$bot),dnn = c("True", "Predicted"))
confusion
# Error rate
1 - sum(diag(confusion))/sum(confusion)
g = roc(bot ~ p, data = train, quiet = FALSE)
plot(g)
g$auc
```