VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

# PROBABILITY AND STATISTICS (CO2013)

## Assignment

# *"Internet Advertisement"*

**Instructor(s):** Nguyễn Tiến Dũng

**Students:** *Phạm Tấn Đức - 2352268*
*Trần Nguyên Khang - 2352505*
*Hoàng Minh Khoa - 2352556*
*Doãn Anh Khôi - 2352601*
*Đinh Tiến - 2353174*

HO CHI MINH CITY, MAY 2025

# Contents

# List of Figures

# List of Tables

# Member list & Workload

| No. | Fullname | Student ID | Problems | % done |
|---|---|---|---|---|
| 1 | Phạm Tấn Đức | 2352268 | Overview & Descriptive statistics | 100 |
| 2 | Trần Nguyên Khang | 2352505 | Data preprocessing | 100 |
| 3 | Hoàng Minh Khoa | 2352556 | Descriptive & Inferential statistics<br>Logistic Regression & Correlation Analysis | 100 |
| 4 | Doãn Anh Khôi | 2352601 | Descriptive & Inferential statistics | 100 |
| 5 | Đinh Tiến | 2353174 | Theoretical basis | 100 |

Table 1: Member list & workload

# 1 Overview

## 1.1 Dataset Description

The Internet has revolutionized how businesses reach potential customers, with online advertisements becoming a cornerstone of digital marketing strategies. Internet advertisements (or "ads") are designed to promote products, services, or brands to a targeted audience through various digital platforms.

The dataset used in the project comprises 3,279 observations and 1,559 variables, each observation represents an individual image extracted from a web page. Each image is labeled either "ad" or "no-ad," serving as the target variable for the classification process.

The dataset can be found at: Internet Advertisements Data Set

## 1.2 Variables Description

The features of the data set can be divided into 9 categories, which are:

| **Category** | **Variable Type** | **Description** |
|---|---|---|
| URL Terms | Binary | Presence of specific terms in the image's URL (e.g., `url*images+buttons`). |
| Original URL Terms | Binary | Presence of terms in the original linking URL (e.g., `origurl*labyrinth`). |
| Anchor URL Terms | Binary | Presence of terms from the anchor text's URL (e.g., `ancurl*search+direct`). |
| Alt Text Terms | Binary | Presence of words in the image's alt attribute (e.g., `alt*your`). |
| Caption Text Terms | Binary | Presence of words in the image's caption (e.g., `caption*and`). |
| Width | Continuous | Width of the image. |
| Height | Continuous | Height of the image. |
| Aspect Ratio | Continuous | Ratio of the image's width to its height. |
| Label (Ad/No-Ad) | Binary | Target variable indicating whether the image is an advertisement or not. |

Table 2: Variable descriptions for the Internet Advertisements Dataset.

The continuous variables—width, height, and aspect ratio—play a more significant role in distinguishing between ad and non-ad instances compared to the binary classification target and the large set of binary features. These binary features, which represent the presence or absence of specific words or phrases in the image URLs, number in the thousands. While they can be grouped or reduced in dimensionality, such transformations often strip away their contextual meaning and reduce their discriminative effectiveness. In contrast, continuous features offer consistent, quantitative variation that captures structural properties of the images, making them more insightful for pattern recognition and classification. Therefore, our report will focus primarily on analyzing these three continuous variables alongside the classification outcome.

# 2 Theoretical Basis

## 2.1 Logistic Regression

### 2.1.1 Definition

Let $\mathbf{x} = (x_1, x_2, \ldots, x_p)^T \in \mathbb{R}^p$ be a vector of predictor variables, and let $y \in \{0, 1\}$ be a binary response variable indicating the class label. The logistic regression model specifies the conditional probability of $y = 1$ given $\mathbf{x}$ as:

$$P(y = 1 \mid \mathbf{x}) = \pi(\mathbf{x}) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p))} = \frac{1}{1 + e^{-\mathbf{x}^T \boldsymbol{\beta}}}$$

where:

- $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T \in \mathbb{R}^{p+1}$ is the vector of regression coefficients (including the intercept),
- $\mathbf{x}^T \boldsymbol{\beta} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$,
- $\pi(\mathbf{x}) \in (0, 1)$ is the predicted probability that the observation belongs to class 1.

### 2.1.2 Estimation via Maximum Likelihood

Given a dataset $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$, we assume $y^{(i)} \sim \text{Bernoulli}(\pi(\mathbf{x}^{(i)}))$. The likelihood function is:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \pi(\mathbf{x}^{(i)})^{y^{(i)}} [1 - \pi(\mathbf{x}^{(i)})]^{1 - y^{(i)}}$$

The log-likelihood is:

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[ y^{(i)} \log \pi(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \log(1 - \pi(\mathbf{x}^{(i)})) \right]$$

To obtain the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$, solve:

$$\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0$$

This is typically done using numerical optimization methods such as Newton-Raphson or Iteratively Reweighted Least Squares (IRLS).

### 2.1.3 Interpretation

Logistic regression models the log-odds (also known as the logit) of the binary response variable as a linear function of the predictors. The model is given by:

$$\log \left( \frac{\pi(x)}{1 - \pi(x)} \right) = x^T \beta$$

where:

- $\pi(x) = P(Y = 1 \mid x)$ is the probability that the outcome $Y$ equals 1 (e.g., the instance is an “ad”),
- $x$ is the vector of input features,

- $\beta$ is the vector of model coefficients.

Each coefficient $\beta_j$ represents the expected change in the **log-odds** of the outcome per one-unit increase in predictor $x_j$, assuming all other variables are held constant:

- If $\beta_j > 0$, then as $x_j$ increases, the odds of $Y = 1$ also increase.
- If $\beta_j < 0$, then as $x_j$ increases, the odds of $Y = 1$ decrease.

Importantly, logistic regression does not produce a binary class directly. Instead, it outputs a predicted probability $\pi(x) \in [0, 1]$. A classification decision can then be made by applying a threshold, as follows:

$$\hat{y} = \begin{cases} 1 & \text{if } \pi(x) \geq p \\ 0 & \text{if } \pi(x) < p \end{cases}$$

In this setup:

- If the predicted probability is greater than or equal to p, we classify the instance as class 1.
- Otherwise, we classify it as class 0.

This probabilistic interpretation allows flexibility in classification and enables confidence estimation for each prediction.

### 2.1.4 Application to the Advertisement Classification Task

In the context of classifying images as either advertisements or non-advertisements, logistic regression provides a natural and effective solution. Given a set of input features $x$, extracted from image metadata or content, the model estimates the probability $\pi(x)$ that the image is an advertisement (class 1).

$$\pi(x) = \mathbb{P}(y = 1 \mid x) = \frac{1}{1 + \exp(-x^\top \beta)}$$

To convert these probabilities into class labels, a decision rule is applied. The most common threshold is 0.5:

$$\hat{y} = \begin{cases} 1 & \text{if } \pi(x) \geq 0.5 \\ 0 & \text{if } \pi(x) < 0.5 \end{cases}$$

If the predicted probability exceeds or equals 0.5, the image is classified as an advertisement; otherwise, it is considered a non-ad. This threshold can be tuned depending on whether we prioritize reducing false positives or false negatives—for example, in a setting where missing an actual ad is more costly than misclassifying a non-ad.

Using logistic regression in this task allows not only for binary classification but also provides interpretable probability estimates, which can be valuable in decision-making systems.

## 2.2 Correlation Matrix

### 2.2.1 Definition

Let $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n\times p}$ represent a dataset with $n$ observations and $p$ numerical variables. The correlation matrix $\mathbf{R} \in \mathbb{R}^{p\times p}$ is defined by:

$$R_{ij} = \rho_{ij} = \frac{\mathrm{Cov}(\mathbf{x}_i, \mathbf{x}_j)}{\sigma_i \sigma_j}$$

where:

- $\mathrm{Cov}(\mathbf{x}_i, \mathbf{x}_j)$ is the covariance,
- $\sigma_i, \sigma_j$ are the standard deviations,
- $\rho_{ij} \in [-1, 1]$ is the Pearson correlation coefficient.

### 2.2.2 Properties

- $R_{ii} = 1$ for all $i$
- $R_{ij} = R_{ji}$
- $\rho_{ij} = 0$ implies no linear correlation
- $|\rho_{ij}|$ close to 1 implies strong linear relationship

### 2.2.3 Mathematical Computation

Given two variables $X$ and $Y$, the Pearson coefficient is:

$$\rho_{XY} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

The empirical correlation matrix can be computed as:

$$\mathbf{R} = \mathbf{D}^{-1}\mathbf{S}\mathbf{D}^{-1}$$

where:

- $\mathbf{S}$: sample covariance matrix,
- $\mathbf{D}$: diagonal matrix of standard deviations.

# 3 Data Preprocessing

## 3.1 Read data

To begin with, the command read.csv() is used to read the data in add.csv into RStudio. Since the unprocessed dataset contains 1559 columns, which can not be fully displayed, our team choose to display column 1 to 15. The data of the first 10 samples are binded with the last 10.
**Code:**

```
# Load the data
ad_data <- read.csv("~/HCMUT/242/XSTK/Assignment/dataset/add.csv")

# Select the columns
cols_to_show <- c(1:15, ncol(ad_data))
# Display first and last 10 rows
rbind(head(ad_data[, cols_to_show], 10),
      tail(ad_data[, cols_to_show], 10))
```

**Output:**

```
        X   X0   X1     X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13  X1558
1       0  125  125      1  1  0  0  0  0  0  0   0   0   0   0    ad.
2       1   57  468 8.2105  1  0  0  0  0  0  0   0   0   0   0    ad.
3       2   33  230 6.9696  1  0  0  0  0  0  0   0   0   0   0    ad.
4       3   60  468    7.8  1  0  0  0  0  0  0   0   0   0   0    ad.
5       4   60  468    7.8  1  0  0  0  0  0  0   0   0   0   0    ad.
6       5   60  468    7.8  1  0  0  0  0  0  0   0   0   0   0    ad.
7       6   59  460 7.7966  1  0  0  0  0  0  0   0   0   0   0    ad.
8       7   60  234    3.9  1  0  0  0  0  0  0   0   0   0   0    ad.
9       8   60  468    7.8  1  0  0  0  0  0  0   0   0   0   0    ad.
10      9   60  468    7.8  1  0  0  0  0  0  0   0   0   0   0    ad.
3270 3269    ?    ?      ?  1  0  0  0  0  0  0   0   0   0   0 nonad.
3271 3270    ?    ?      ?  1  0  0  0  0  0  0   0   0   0   0 nonad.
3272 3271    ?    ?      ?  1  0  0  0  0  0  0   0   0   0   0 nonad.
3273 3272  106  110 1.0377  1  0  0  0  0  0  0   0   0   0   0 nonad.
3274 3273   30   30      1  0  0  0  0  0  0  0   0   0   0   0 nonad.
3275 3274  170   94 0.5529  0  0  0  0  0  0  0   0   0   0   0 nonad.
3276 3275  101  140 1.3861  1  0  0  0  0  0  0   0   0   0   0 nonad.
3277 3276   23  120 5.2173  1  0  0  0  0  0  0   0   0   0   0 nonad.
3278 3277    ?    ?      ?  1  0  0  0  0  0  0   0   0   0   0 nonad.
3279 3278   40   40      1  1  0  0  0  0  0  0   0   0   0   0 nonad.
```

## 3.2 Clean the data, change variable formats

First, we remove Column 1, which contains only sample identifiers and is not relevant for analysis. Columns 4 through 1558 are also excluded, as they do not contribute meaningful information to the objectives of this project. The variables X0, X1, and X2, representing height, width, and ratio respectively, are converted from character strings to numeric format to enable quantitative analysis. Any unknown values denoted by the character '?' are automatically converted to NA to standardize missing data representation. Additionally, the categorical values "ad." and "nonad." in variable X1558 are recoded from character strings into binary format, with "ad." mapped to 1 and "nonad." mapped to 0.

Additionally, for improved readability, our team has renamed the variables 'X0', 'X1', 'X2', and 'X1558' to 'height', 'width', 'ratio', and 'target' respectively.

**Code:**

```
# Remove the first column (No.)
ad_data <- ad_data[, -1]

# Remove col 4 to 1558
ad_data <- ad_data[, -c(4:1558)]

# Convert X0, X1, X2 from character to numeric
ad_data$X0 <- as.numeric(ad_data$X0)
ad_data$X1 <- as.numeric(ad_data$X1)
ad_data$X2 <- as.numeric(ad_data$X2)

# Rename columns
```

```
colnames(ad_data)[1] <- "height"
colnames(ad_data)[2] <- "width"
colnames(ad_data)[3] <- "ratio"
colnames(ad_data)[4] <- "target"

# Convert target to binary
ad_data$target <- ifelse(ad_data$target == "ad.", 1, 0)

# Display first and last 10 rows
rbind(head(ad_data, 10),
      tail(ad_data, 10))
```

**Output:**

```
     height width  ratio target
1       125   125 1.0000      1
2        57   468 8.2105      1
3        33   230 6.9696      1
4        60   468 7.8000      1
5        60   468 7.8000      1
6        60   468 7.8000      1
7        59   460 7.7966      1
8        60   234 3.9000      1
9        60   468 7.8000      1
10       60   468 7.8000      1
3270     NA    NA     NA      0
3271     NA    NA     NA      0
3272     NA    NA     NA      0
3273    106   110 1.0377      0
3274     30    30 1.0000      0
3275    170    94 0.5529      0
3276    101   140 1.3861      0
3277     23   120 5.2173      0
3278     NA    NA     NA      0
3279     40    40 1.0000      0
```

## 3.3 Check for missing values

Since the missing values in height, width, and ratio have already been converted to NA, we use colSums() and colMeans() to calculate the number and percentage of missing values, respectively.

**Code:**

```

# Count number of NA
na_counts <- colSums(is.na(ad_data))
# Percentage of NA for every col
na_percentage <- colMeans(is.na(ad_data)) * 100
na_summary <- data.frame(
  NA_Count = na_counts,
  NA_Percent = round(na_percentage, 2)
)
print(na_summary)

# Handle missing data by imputation
# Replace missing values with median
ad_data$height[is.na(ad_data$height)] <- median(ad_data$height, na.rm = TRUE)
ad_data$width[is.na(ad_data$width)] <- median(ad_data$width, na.rm = TRUE)

# Recalculate Ratio
update_ratio <- function(df) {
  df$ratio[is.na(df$ratio) & !is.na(df$height) & !is.na(df$width)] <- df$height[is.na(df$ratio) & !is.na(df$height) & !is.na(df$width)] / df$width[is.na(df$ratio) & !is.na(df$height) & !is.na(df$width)]
  return(df)
}

# Use update_ratio to replace missing values
ad_data <- update_ratio(ad_data)
```

**Output:**

```
           NA_Count  NA_Percent
height       903       27.54
width        901       27.48
ratio        910       27.75
target         0        0.00
```

The result shows that approximately 28% of the values are missing in each of these continuous attributes, which is consistent with the author's statement. Since the percentage of missing data is considerable, our team choose to replace missing values of 'height' and 'width' with the median instead of removing the observation. With the imputed values, the column 'ratio' is then recalculated to maintain data consistency.

# 4 Descriptive Statistics

## 4.1 Summary Statistics

The dataset is categorized into two main types of variables: continuous variables and discrete variables. The continuous variables are "Height", "Width",and "Aspect Ratio". In order to get a deeper understanding of the data set, we calculate the descriptive statistical values: mean, standard deviation, median, first quantile, third quantile, minimum value, and maximum value using R:
**Code:**

```
# Select numeric variables (excluding the binary target)
numeric_data <- ad_data[, c(1, 2, 3)]

# Calculate summary statistics with more metrics
summary_stats <- sapply(numeric_data, function(x) {
  c(Mean = mean(x),
    SD = sd(x),
    Variance = var(x),
    Min = min(x),
    Q1 = quantile(x, 0.25),
    Median = median(x),
    Q3 = quantile(x, 0.75),
    Max = max(x),
    IQR = IQR(x))
})
# Transpose and round for better readability
stats_table <- t(as.data.frame(summary_stats))
round(stats_table, 2)
```

**Output:**

```
         Mean     SD Variance Min  Q1.25% Median  Q3.75% Max   IQR
height  60.44  47.06  2215.08   1   32.50   51.0   61.0 640 28.50
width  142.89 112.56 12670.79   1   90.00  110.0  144.0 640 54.00
ratio    3.41   5.20    27.04   0    1.28    2.1    3.9  60  2.62
```

## 4.2 Data Visualization

In this section, we will explore the visual representations of the data set to obtain a more comprehensive and intuitive understanding of the data set. Through various graphs and plots, this section aims to reveal patterns, trends, distributions, and relationships in the data.

### 4.2.1 Histogram of Continuous Variables

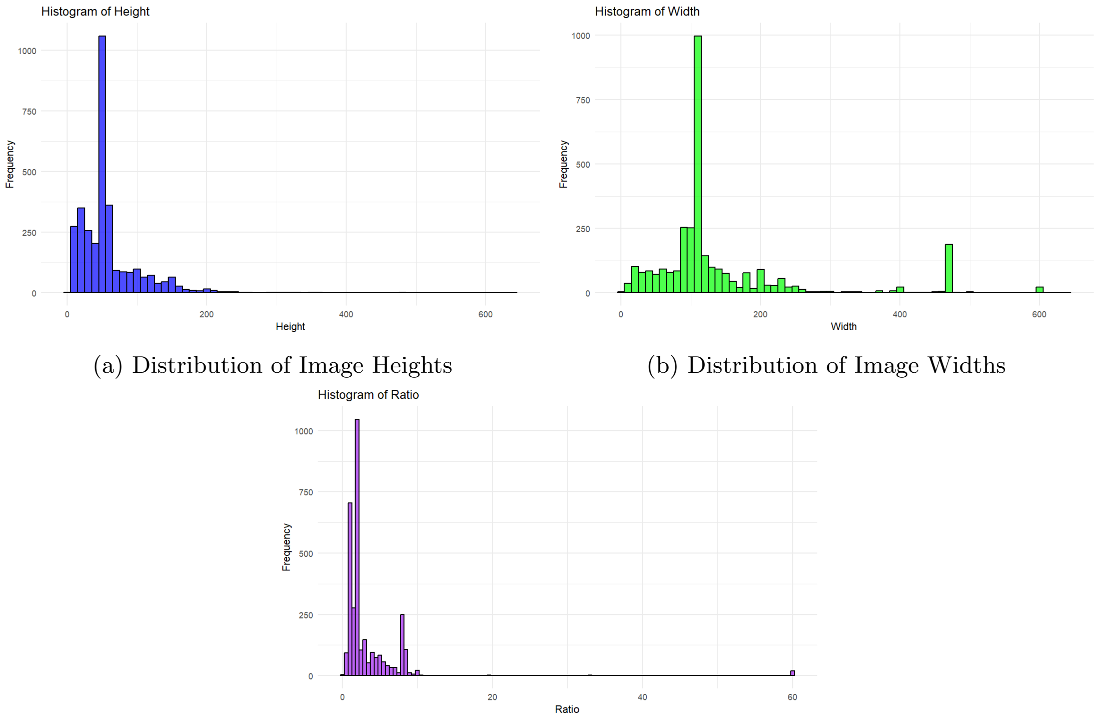
The purpose of this section is to visualize the distributions of key features (height, width, and ratio) in the dataset through histograms. By plotting these histograms, we can better understand the spread, central tendency, skewness, and potential outliers in the data. This is useful when applying models like logistic regression, as certain algorithms may assume that data is normally distributed, or at least want to avoid highly skewed data.

We utilized the "ggplot2" package in R to generate comprehensive visualizations of the continuous variables, specifically height, width, and aspect ratio, using density histograms.

**Code:**

```r
# Histogram for height
ggplot(ad_data, aes(x = height)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Height", x = "Height", y = "Frequency") +
  theme_minimal()

# Histogram for width
ggplot(ad_data, aes(x = width)) +
  geom_histogram(binwidth = 10, fill = "green", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Width", x = "Width", y = "Frequency") +
  theme_minimal()

# Histogram for ratio
ggplot(ad_data, aes(x = ratio)) +
  geom_histogram(binwidth = 0.5, fill = "purple", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Ratio", x = "Ratio", y = "Frequency") +
  theme_minimal()
```

**Output:**



(a) Distribution of Image Heights

(b) Distribution of Image Widths

(c) Distribution of Image Aspect Ratios

Figure 1: Histogram Representations of Key Image Attributes

When examining the three histograms of height, width, and aspect ratio, it is evident that all of them show right-skewed distributions. The peaks of these histograms are all located towards the left side of the plots, indicating that the majority of the data values are concentrated in the lower ranges, whilst their tails extend towards the higher values, and the mean, or average, is greater than the median

The aspect ratio histogram shows an especially sharp peak near zero, with a long tail extending to the right, indicating that most images are significantly wider than they are tall. Similarly, the height and width histograms also show strong right skewness, though the width histogram appears to be more spread out and bimodal, suggesting a greater diversity in object dimensions.

### 4.2.2 Stripchart

To gain a deeper understanding of how continuous variables relate to the target variable, an effective approach involves analyzing their interactions and dependencies. By examining relationships between different numerical features, we can uncover patterns, correlations, and potential influences that contribute to predicting or explaining the target variable.

**Code:**

```
# Set up 3 plots in a column layout
par(mfrow = c(3, 1), mar = c(5, 5, 4, 2))

# Strip Plot: Height vs Ad/NoAd
stripchart(height ~ target,
           data = ad_data,
           horizontal = TRUE,
           method = "jitter",
           pch = 19,
           col = rgb(0.2, 0.4, 0.8, 0.5),  # Semi-transparent blue
           main = "Height Distribution by Ad Status",
           ylab = "Ad Status (0 = Non-Ad, 1 = Ad)",
           xlab = "Height")

# Strip Plot: Width vs Ad/NoAd
stripchart(width ~ target,
           data = ad_data,
           horizontal = TRUE,
           method = "jitter",
           pch = 19,
           col = rgb(0.1, 0.7, 0.1, 0.5),  # Semi-transparent green
           main = "Width Distribution by Ad Status",
           ylab = "Ad Status (0 = Non-Ad, 1 = Ad)",
           xlab = "Width")

# Strip Plot: Ratio vs Ad/NoAd
stripchart(ratio ~ target,
           data = ad_data,
           horizontal = TRUE,
           method = "jitter",
           pch = 19,
           col = rgb(0.9, 0.2, 0.2, 0.5),  # Semi-transparent red
           main = "Aspect Ratio Distribution by Ad Status",
           ylab = "Ad Status (0 = Non-Ad, 1 = Ad)",
           xlab = "Aspect Ratio")

# Reset layout
par(mfrow = c(1, 1))
```

**Output:**

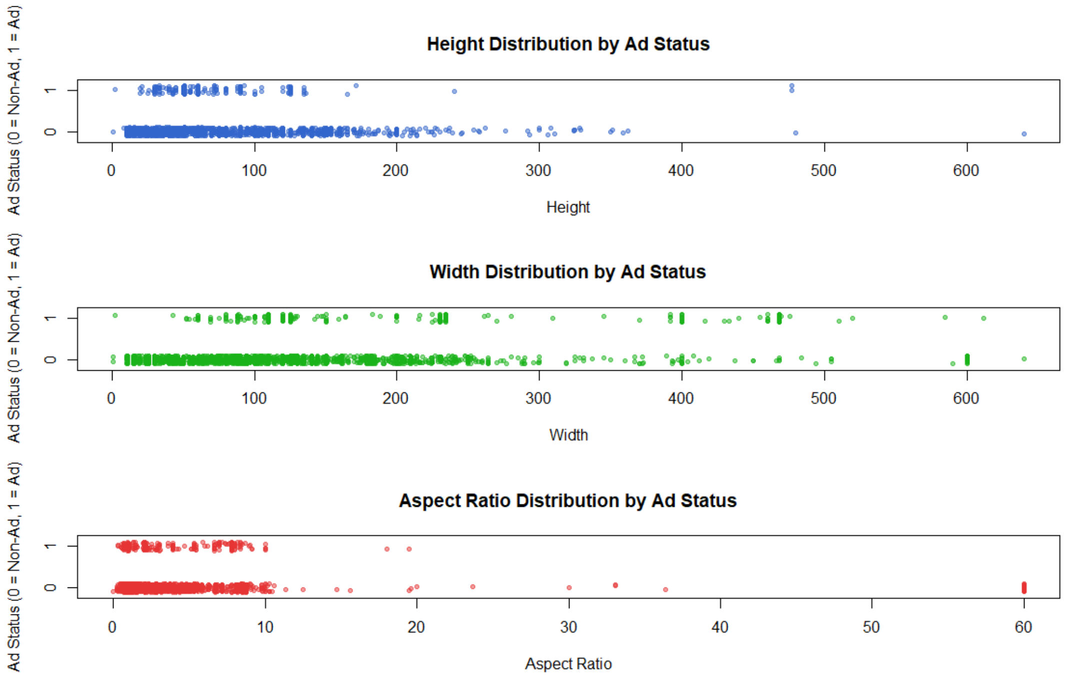


Figure 2: Strip chart of Height, Width, and Aspect Ratio to Target

The Height distribution strip chart represents the height distribution of images by ad status. Observing the upper line corresponding to advertisements (target = 1), it is evident that ad images display a concentrated distribution, with most heights falling within a moderate range below 200 pixels. In contrast, non-advertisement images (target = 0), represented by the lower line, show a broader distribution with a significant number of images extending beyond 200 pixels, and some even exceeding 600 pixels. This suggests that advertisements tend to maintain more uniform heights, possibly adhering to specific design standards, while non-advertisement images are more variable in their height.

The width distribution strip chart reveals a clearer distinction between advertisements and non-advertisements. Non-advertisements are primarily concentrated below 300 units, whereas advertisements have a broader spread, with significant occurrences between 300 and 600 units. This could imply that advertisements are designed in larger formats to accommodate banners, promotional displays, or other visual needs. The wider range of widths in advertisements may be an important feature for the logistic regression model to consider when predicting ad status.

For the Aspect Ratio distribution strip chart, advertisements show a highly clustered distribution at lower aspect ratios, primarily below 10, indicating that ad images often have nearly square or slightly elongated dimensions. Non-advertisements, on the other hand, exhibit a wider range of aspect ratios, with several outliers exceeding 60. This highlights that advertisements tend to conform to standard aspect ratios, likely to maintain aesthetic and functional consistency, whereas non-advertisements display significant diversity in their proportions. The distinct separation in aspect ratios between the two classes suggests that this feature may be particularly useful for distinguishing advertisements from non-advertisements in a predictive model.

### 4.2.3 Bar plot

The bar plot visualization provides an overview of the distribution of the target variable in the dataset

**Code:**

```
# Barplot for target
ggplot(ad_data, aes(x = factor(target), fill = factor(target))) +
  geom_bar() +
  scale_fill_manual(values = c("blue", "orange")) +
  labs(title = "Barplot for Target", x = "Target (1=Ad, 0=Non-Ad)", y = "Count") +
  theme_minimal()
```

**Output:**

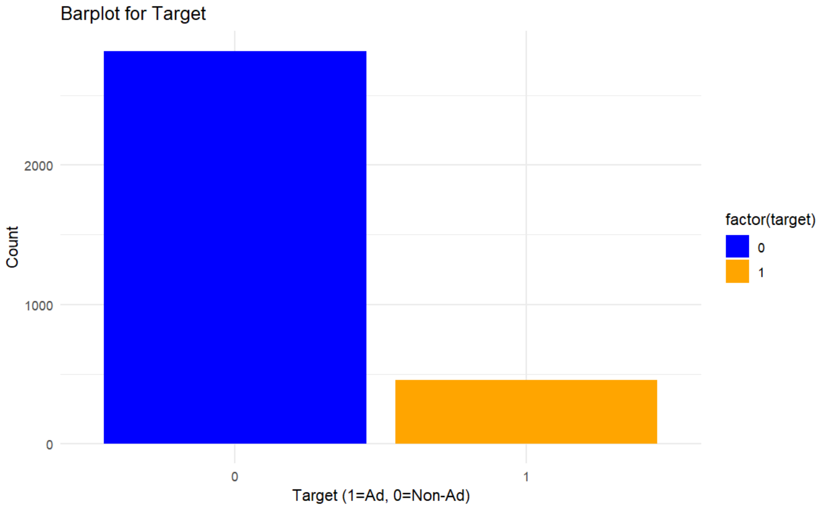


Figure 3: Frequency of Targets

The graph clearly shows that ad-labeled images are significantly fewer than non-ad images. This imbalance suggests that ads make up only a small portion of the dataset.

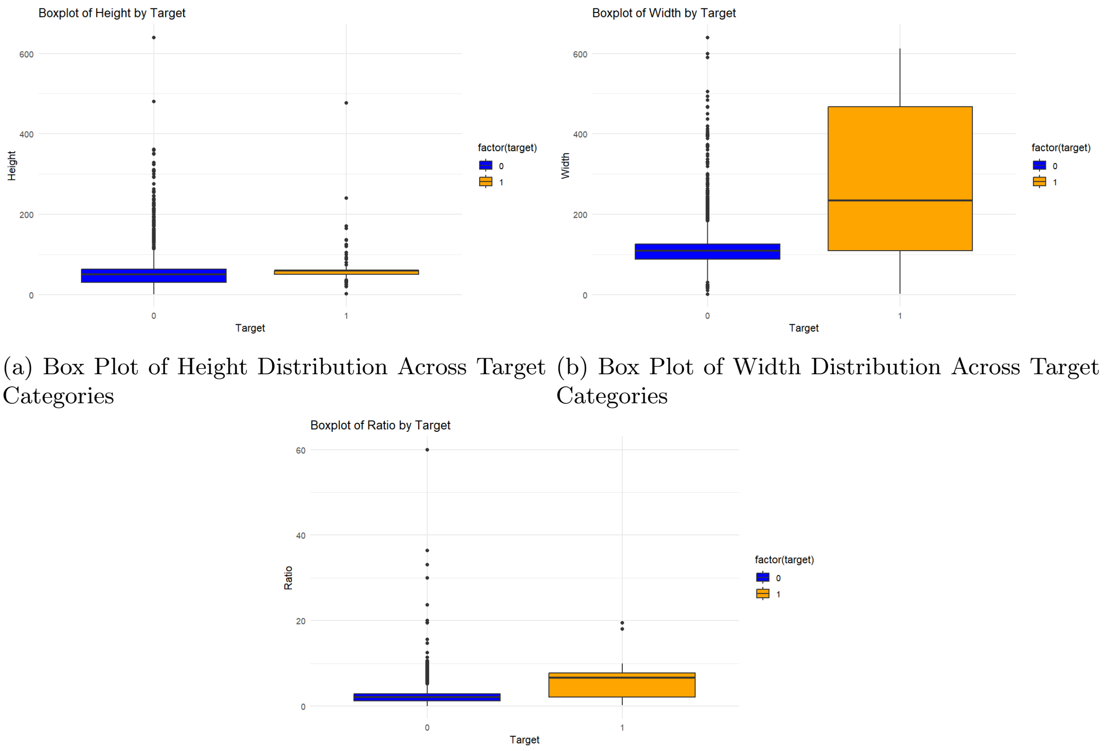
### 4.2.4 Box plot

Box plots are the ideal choice for visualizing the distribution of data, highlighting measures of central tendency, and identifying differences in quartiles, mean, and outliers. They provide a clear representation of how values spread across a dataset, making it easier to compare variations and detect anomalies.

**Code:**

```
    # Boxplot for height by target
ggplot(ad_data, aes(x = factor(target), y = height, fill = factor(target))) +
  geom_boxplot() +
  scale_fill_manual(values = c("blue", "orange")) +
  labs(title = "Boxplot of Height by Target", x = "Target", y = "Height") +
  theme_minimal()

# Boxplot for width by target
ggplot(ad_data, aes(x = factor(target), y = width, fill = factor(target))) +
  geom_boxplot() +
  scale_fill_manual(values = c("blue", "orange")) +
  labs(title = "Boxplot of Width by Target", x = "Target", y = "Width") +
  theme_minimal()

# Boxplot for ratio by target
ggplot(ad_data, aes(x = factor(target), y = ratio, fill = factor(target))) +
  geom_boxplot() +
  scale_fill_manual(values = c("blue", "orange")) +
  labs(title = "Boxplot of Ratio by Target", x = "Target", y = "Ratio") +
  theme_minimal()
```

**Output:**



(a) Box Plot of Height Distribution Across Target Categories

(b) Box Plot of Width Distribution Across Target Categories

(c) Box Plot of Aspect Ratio Distribution Across Target Categories

Figure 4: Box Plots Showing the Distribution of Key Image Features

The height distribution shows a clear separation Target 0 and Target 1.Target 1 has a narrower range compared to the Target 0, indicating that Target 1 tends to have smaller height range. The outliers in both categories suggest that height is a variable with variability across both targets, but the tighter distribution in Target 1 makes it a potential distinguishing feature. This finding indicates that height could contribute meaningfully to identifying instances belonging to Target 1.

Width demonstrates significant differences between Target 0 and Target 1.Target 1 exhibits a much broader range and higher median width value compared to Target 0. This stark contrast highlights width as a highly discriminative feature, especially for distinguishing Target 1, which has a more varied and larger width. These characteristics suggest that width could play a pivotal role in the classification process, offering strong predictive value.

The aspect ratio provides a composite measure of both height and width, and the box plots reveal differences between Target 0 and Target 1. Target 1 shows a broader range and higher median value compared to Target 0. This indicates that Target 1 is associated with larger aspect ratios, making this feature useful for capturing proportional relationships between height and width. The variability in aspect ratio suggests its potential for identifying subtle structural differences between the two target categories.

On the other hand, the distinct clustering of width and aspect ratio in ad-labeled images indicates that ads are designed with larger sizes and preferred structural formats. This differentiation implies that width and aspect ratio may serve as strong distinguishing features between "ad." and "noad." targets.

## 4.3 Correlation Matrix

To further examine the linear relationships among the continuous predictors, we computed the Pearson correlation matrix using the `cor()` function. The result was visualized using the `corrplot` package in R, which provided a graphical summary of the correlation coefficients between `height`, `width`, and `ratio` (Figure 5).

**Code:**

```r
# Calculate the correlation matrix between the continuous variables 'height', '
    width', 'ratio'
library(corrplot)
cor_matrix <- cor(ad_data[, c("height", "width", "ratio")], use = "complete.obs")

# Draw the correlation matrix plot
corrplot(cor_matrix, method = "circle", type = "upper",
         addCoef.col = "black", tl.col = "black", number.cex = 0.8,
title = "Correlation Matrix", mar = c(0,0,1,0))
```

**Output:**

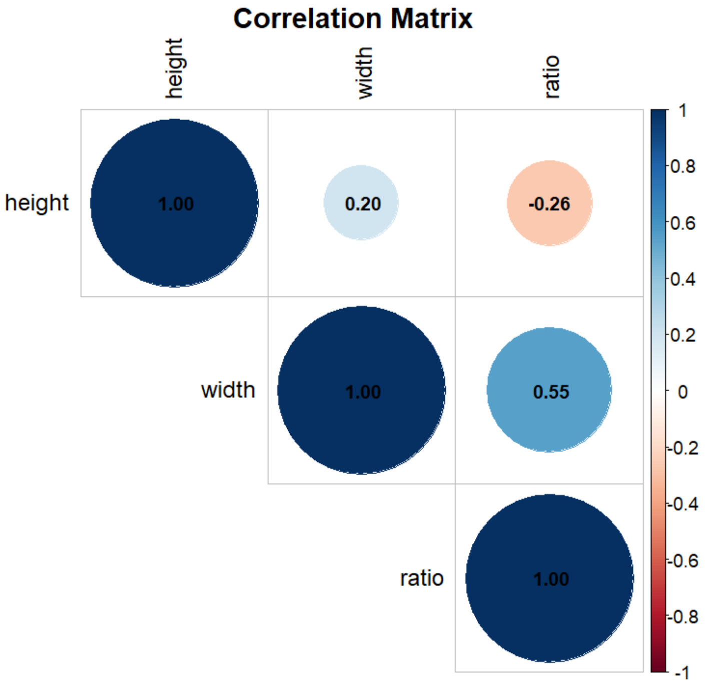


Figure 5: Correlation Matrix Plot for Height, Width, and Ratio

The correlogram reveals that width and aspect ratio have a moderate positive correlation (0.55), indicating that as width increases, the aspect ratio (width divided by height) also tends to increase. This relationship is expected given that width directly influences the aspect ratio. In contrast, height shows only a weak positive correlation with width (0.20) and a weak negative correlation with aspect ratio (-0.26). These values suggest that taller objects are only slightly more likely to be wider and tend to have lower aspect ratios, meaning they are relatively narrower.

Overall, these findings imply that while width and aspect ratio share some linear dependency, height remains largely independent from the other two features. This low redundancy suggests that each variable would offer unique contributions when included in the predictive model.

# 5 Logistic Regression analysis

To explore the relationship between the binary outcome variable `target` and the continuous predictors `height`, `width`, and `ratio`, we performed a series of logistic regression analyses using the `glm()` function in R with the binomial family. We fitted separate logistic regression models for each predictor to assess their individual influence on the probability of the target outcome. The relationship is expressed as a sigmoid curve that maps the input values to probabilities ranging from 0 to 1.

The following models were constructed:

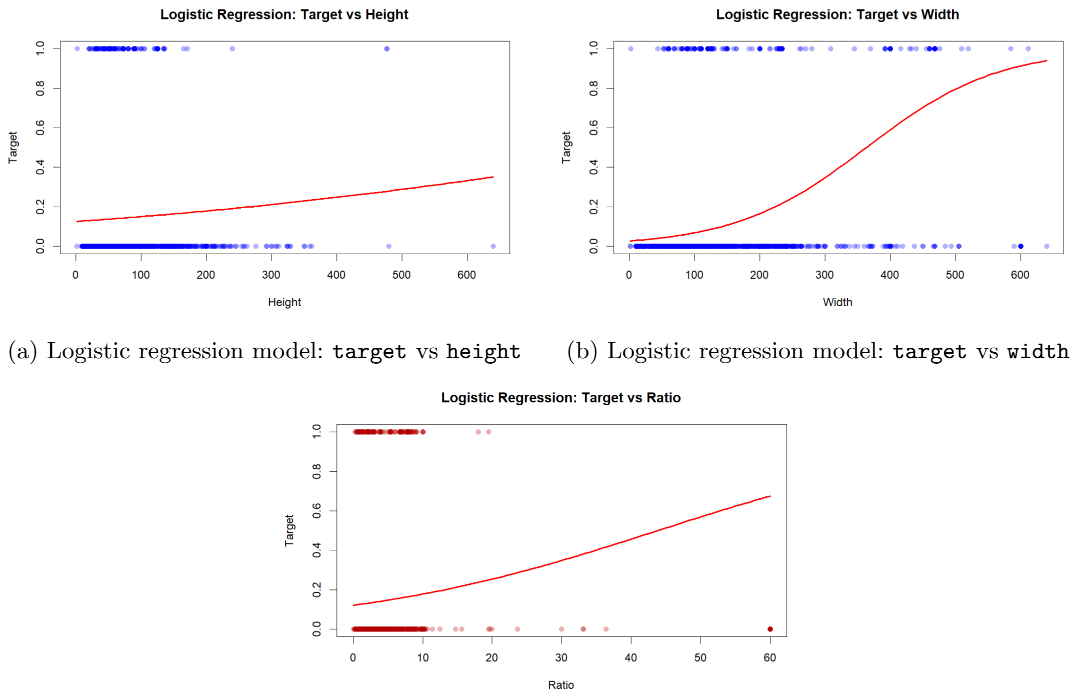
- **Height model:** `model_height <- glm(target █eight, data = ad_data, family = "binomial")`
- **Width model:** `model_width <- glm(target █idth, data = ad_data, family = "binomial")`
- **Ratio model:** `model_ratio <- glm(target █atio, data = ad_data, family = "binomial")`

For each model, a scatter plot of the predictor variable against the binary target was generated, with the logistic regression curve overlaid in red to visualize the modeled probability. These visualizations are presented in Figures 6a, 6b, and 6c.

**Code:**

```r
# Logistic regression model for height
model_height <- glm(target ~ height, data = ad_data, family = "binomial")

# Draw distribution chart between 'height' and 'target'
# then draw logistic curve
plot(ad_data$height, ad_data$target,
     xlab = "Height", ylab = "Target",
     main = "Logistic Regression: Target vs Height",
     pch = 16, col = rgb(0, 0, 1, 0.3))  # Distribution chart

curve(predict(model_height, newdata = data.frame(height = x), type = "response"),
      add = TRUE, col = "red", lwd = 2)  # logistic curve

# Logistic regression model for width
model_width <- glm(target ~ width, data = ad_data, family = "binomial")

plot(ad_data$width, ad_data$target,
     xlab = "Width", ylab = "Target",
     main = "Logistic Regression: Target vs Width",
     pch = 16, col = rgb(0, 0, 1, 0.3))

curve(predict(model_width, newdata = data.frame(width = x), type = "response"),
      add = TRUE, col = "red", lwd = 2)

# Logistic regression model for ratio
model_ratio <- glm(target ~ ratio, data = ad_data, family = "binomial")

plot(ad_data$ratio, ad_data$target,
     xlab = "Ratio", ylab = "Target",
     main = "Logistic Regression: Target vs Ratio",
     pch = 16, col = rgb(0.7, 0, 0, 0.3))

curve(predict(model_ratio, newdata = data.frame(ratio = x), type = "response"),
      add = TRUE, col = "red", lwd = 2)
```

**Output:**



(a) Logistic regression model: `target` vs `height`

(b) Logistic regression model: `target` vs `width`

(c) Logistic regression model: `target` vs `ratio`

The logistic regression model for height suggests a weak relationship between height and advertisement status. The scatter plot indicates that data points are dispersed, with no clear separation between advertisements and non-advertisements. The logistic curve is relatively flat, implying that height alone does not significantly impact the probability of an image being an advertisement. This means height might not be a strong predictor in our model, and additional features may be needed to improve classification accuracy.

In contrast, the logistic regression model for width displays a more noticeable pattern. The scatter plot shows a clearer distinction between advertisement and non-advertisement images, with width having a stronger influence on ad classification. The logistic curve is steeper, indicating that as width increases, the probability of an image being an advertisement rises more predictably. This suggests that width could be an important factor for determining advertisement status, making it more relevant to the logistic regression model than height.

The logistic regression model for aspect ratio presents a moderate relationship with the target variable. The scatter plot displays some differentiation between advertisements and non-advertisements, though not as distinctly as width. The logistic curve is more pronounced than in the height model but less extreme than in the width model. This indicates that ratio plays a role in distinguishing advertisements but may not be the most dominant predictor. However, combining ratio with width could enhance classification performance, contributing to a more effective predictive model.

# 6 Inferential Statistics

## 6.1 Logistic Regression Models

To assess the influence of predictors width and ratio on the binary outcome target, logistic regression models were built and evaluated using training and testing datasets (70%-30% split).
**Code:**

```
# Divide the dataset into training set and testing set
# 70% - 30%
sample_index <- sample(1:nrow(ad_data), size = 0.7 * nrow(ad_data))
train_data <- ad_data[sample_index, ]
test_data <- ad_data[-sample_index, ]
```

Two separate models were considered:

- Model 1: Logistic regression width height and width as predictors.
- Model 2: Logistic regression with width as the sole predictor.
- Model 3: Logistic regression with ratio as the sole predictor.

The models were fitted using the glm() function in R with the binomial family and training dataset (70% of the original data). The summary() function provides model details, such as:

- Coefficients: How each variable contributes to predicting target.
- Significance: Whether the coefficients are statistically significant.
- AIC: A metric for comparing model fit (lower AIC indicates a better fit).

### 6.1.1 Model 1

**Code:**

```
# Logistic regression model with ’Height’ and ’Width’
model_hw <- glm(target ~ height + width, data = train_data, family = binomial)
summary(model_hw)
```

**Result:**

```
Call:
glm(formula = target ~ height + width, family = binomial, data = train_data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.3838989  0.1453824 -23.276   <2e-16 ***
height      -0.0032037  0.0015195  -2.108    0.035 *
width        0.0101898  0.0005167  19.719   <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: 1953.2  on 2294  degrees of freedom
Residual deviance: 1436.8  on 2292  degrees of freedom
AIC: 1442.8

Number of Fisher Scoring iterations: 5
```

This model evaluates the combined influence of height and width on the target variable. Both predictors are statistically significant, with width having a stronger effect ($p < 2e\text{-}16$) compared to height ($p = 0.035$). The positive coefficient for width suggests that an increase in width increases the likelihood of the target outcome, while the negative coefficient for height indicates a slight decrease in the likelihood as height increases. This model has the lowest AIC (1442.8), making it the best fit among the three models.

### 6.1.2 Model 2

**Code:**

```
# Logistic regression model with only 'Width'
model_hw2 <- glm(target ~ width, data = train_data, family = binomial)
summary(model\_hw2)
```

**Result:**

```
Call:
glm(formula = target ~ width, family = binomial, data = train_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.562210   0.122360  -29.11   <2e-16 ***
width        0.010025   0.000506   19.81   <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: 1953.2  on 2294  degrees of freedom
Residual deviance: 1441.5  on 2293  degrees of freedom
AIC: 1445.5

Number of Fisher Scoring iterations: 5
```

This simpler model includes only width as a predictor. The results show that width is a highly significant factor ($p < 2e\text{-}16$) in predicting the target variable. Its positive coefficient confirms that larger values of width increase the likelihood of the target outcome. Although the AIC (1445.5) is slightly higher than Model 1, the model performs nearly as well, offering a balance between simplicity and predictive power.

### 6.1.3 Model 3

**Code:**

```
# Logistic regression model with 'Ratio'
model_ratio <- glm(target ~ ratio, data = train_data, family = binomial)
summary(model_ratio)
```

**Result:**

```
Call:
glm(formula = target ~ ratio, family = binomial, data = train_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.901846   0.068581 -27.731  < 2e-16 ***
ratio        0.046728   0.008928   5.234 1.66e-07 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1
```

```

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1953.2  on 2294  degrees of freedom
Residual deviance: 1923.6  on 2293  degrees of freedom
AIC: 1927.6

Number of Fisher Scoring iterations: 4
```

This model examines the effect of ratio on the target variable. The predictor is significant (p = 1.66e-07), with a positive coefficient indicating that higher ratio values increase the likelihood of the target outcome. However, the model has a much higher AIC (1927.6) compared to the others, indicating a poorer fit. This suggests that ratio alone is not as effective as width or the combination of height and width in predicting the target variable.

## 6.2 Model evaluation

Only two of the three fitted models (width and ratio) were analyzed in detail to simplify comparison and focus on key predictors. The combined model with height + width was excluded from ROC and AUC evaluation to keep the analysis clear and interpretable, since height had weaker influence and added little improvement.

### 6.2.1 Evaluation criteria

Each model was evaluated using the testing dataset (30% of the original data) based on the following criteria:

- Accuracy: The proportion of correct predictions (based on a probability threshold of 0.5).
- Confusion Matrix: Displays the relationship between predicted and actual classifications.
- Akaike Information Criterion (AIC): A measure of model fit, with lower values indicating a better model.
- ROC Curve and AUC (Area Under the Curve) Assess the model's ability to discriminate between classes. Higher AUC values suggest better model performance.

**Code:**

```r
# Function to evaluate the model
evaluate_model <- function(model, data, label) {
  pred_prob <- predict(model, newdata = data, type = "response")
  pred_class <- ifelse(pred_prob > 0.5, 1, 0)

  actual <- data$target
  accuracy <- mean(pred_class == actual)

  cat(paste0(label, ":\n"))
  cat("Accuracy: ", round(accuracy, 4), "\n")
  cat("Confusion Matrix:\n")
  print(table(Predicted = pred_class, Actual = actual))
  cat("\n")
}

# Evaluate the model
evaluate_model(model_hw2, test_data, "Model 1 (width)")
evaluate_model(model_ratio, test_data, "Model 2 (ratio)")

```

```r
# Print the AIC of models
cat("AIC Model 1 ( width): ", AIC(model_hw2), "\n")
cat("AIC Model 2 (ratio): ", AIC(model_ratio), "\n")
```

**Result:**

```
Model 1 (width):
Accuracy:  0.9207
Confusion Matrix:
          Actual
Predicted   0   1
        0 856  61
        1  17  50

Model 2 (ratio):
Accuracy:  0.8821
Confusion Matrix:
          Actual
Predicted   0   1
        0 868 111
        1   5   0

AIC Model 1 (width):  1445.47
AIC Model 2 (ratio):  1927.635
```

The evaluation of Model 1 (width) and Model 2 (ratio) provides insights into their predictive performance and fit. Model 1, which uses width as the sole predictor, achieves an accuracy of 92.07%, with a relatively balanced confusion matrix: it correctly predicts 856 true negatives and 50 true positives, with 61 false positives and 17 false negatives. This indicates that the model is effective at distinguishing between the two target classes. Additionally, the AIC value for Model 1 is 1445.47, suggesting a good balance between model fit and complexity.

In contrast, Model 2, which uses ratio as the sole predictor, has a lower accuracy of 88.21%. Its confusion matrix reveals poorer classification of true positives, as it predicts no instances of the positive class correctly (0 true positives) and has 5 false negatives, alongside 868 true negatives and 111 false positives. This suggests that the ratio variable alone struggles to predict the target class effectively. Furthermore, the AIC for Model 2 is substantially higher at 1927.635, indicating a worse overall model fit compared to Model 1.

In summary, Model 1 (width) outperforms Model 2 (ratio) in terms of accuracy, AIC, and classification performance, suggesting that width is a stronger standalone predictor of the target variable than ratio.

### 6.2.2 Comparative evaluation

To evaluate and compare the discriminatory ability of two logistic regression models—one using width (Model 1) and the other using ratio (Model 2)—we analyzed their ROC (Receiver Operating Characteristic) curves and corresponding AUC (Area Under the Curve) values. Predicted probabilities for the test dataset were obtained using the predict() function for both models. ROC curves were generated to illustrate the trade-off between the True Positive Rate (Sensitivity) and the False Positive Rate (1 - Specificity) across various probability thresholds. AUC values were computed to quantify the overall performance of each model, with higher AUC values indicating better discrimination. A visual comparison was presented, where the blue curve represented Model 1 (width) and the red curve represented Model 2 (ratio), allowing for a clear contrast in their predictive capabilities.

**Code:**

```r
library(pROC)
```

```
# Prediction on the test dataset
prob_hw <- predict(model_hw2, newdata = test_data, type = "response")
prob_ratio <- predict(model_ratio, newdata = test_data, type = "response")

# Plot ROC
roc_hw <- roc(test_data$target, prob_hw)
roc_ratio <- roc(test_data$target, prob_ratio)

# Plot comparison chart
plot(roc_hw, col = "blue", lwd = 2, main = "ROC Curve Comparison")
lines(roc_ratio, col = "red", lwd = 2)
legend("bottomright", legend = c("Model 1: width", "Model 2: ratio"),
       col = c("blue", "red"), lwd = 2)

# Print AUC
cat("AUC Model 1 (width): ", auc(roc_hw), "\n")
cat("AUC Model 2 (ratio): ", auc(roc_ratio), "\n")
```

**Result:**

```
AUC Model 1 (width):  0.7451936
AUC Model 2 (ratio):  0.645496
```

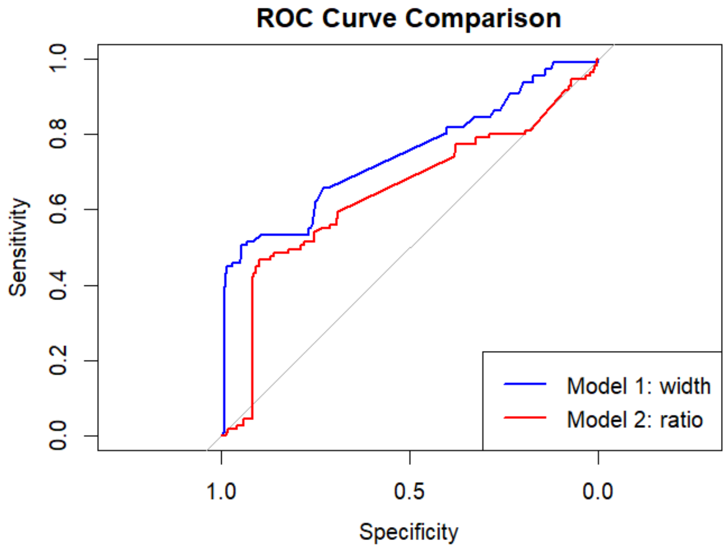


Figure 7: ROC curve comparison of Model 1 and Model 2

The ROC (Receiver Operating Characteristic) curve comparison visually evaluates the performance of two logistic regression models: Model 1 (using width) and Model 2 (using ratio). The blue curve represents Model 1, while the red curve represents Model 2. These curves plot Sensitivity (True Positive Rate) against 1-Specificity (False Positive Rate) across varying probability thresholds. The diagonal grey line serves as a reference, indicating random guessing.

From the plot, it is evident that Model 1 outperforms Model 2, as its curve consistently lies above the curve for Model 2, particularly in areas of higher Sensitivity. This suggests that Model 1 has better discriminatory power for correctly distinguishing between the two target classes.

The AUC (Area Under the Curve) values, which quantify overall model performance, further confirm this observation. Model 1 achieves a higher AUC compared to Model 2, indicating that it is more effective in predicting the target variable. The comparative analysis highlights the superiority of using width as a predictor over the ratio variable in this context.

# 7 Summary

All in all, in line with the requirements of this project, we have made every effort to thoroughly analyze and evaluate the factors influencing the classification of advertisements in the dataset. By leveraging the logistic regression technique and employing R programming as our primary tool, we have effectively processed, analyzed, and visualized the data to draw meaningful insights. The steps involved data preprocessing, feature selection, model development, and evaluation, all aimed at identifying the predictors most strongly associated with the target variable.

During the research, statistical methods such as accuracy analysis, AIC comparison, confusion matrix evaluation, and ROC-AUC analysis were applied to assess model performance. Our analysis led us to the conclusion that the width variable is the most significant predictor, achieving high accuracy and AUC scores in classification tasks. Other features, such as ratio, displayed weaker predictive power but still contributed to understanding the dataset's dynamics.

What is more, through this project, we have honed our skills in programming, statistical modeling, and data visualization. By exploring multiple models and evaluating them against a testing dataset, we have enhanced our understanding of logistic regression and its application in real-world scenarios. The process of building, evaluating, and comparing models has equipped us with the ability to approach complex problems systematically and make data-driven decisions.

Lastly, we extend our heartfelt gratitude to Dr.Nguyen Tien Dung, who provided invaluable guidance and feedback throughout this project. While we acknowledge that, as learners, our analysis may not be perfect, we have put our utmost effort into applying the knowledge gained and conducting this project with diligence and collaboration. This experience has not only deepened our understanding of statistical modeling but also strengthened our teamwork and problem-solving skills, preparing us for future challenges.

# 8 Source code

```r
# Load the data
ad_data <- read.csv("~/HCMUT/242/XSTK/Assignment/dataset/add.csv")

# Select the columns
cols_to_show <- c(1:15, ncol(ad_data))
# Display first and last 10 rows
rbind(head(ad_data[, cols_to_show], 10),
      tail(ad_data[, cols_to_show], 10))

# Part 1: Clean the data

# Remove the first column (No.)
ad_data <- ad_data[, -1]

# Remove col 4 to 1558
ad_data <- ad_data[, -c(4:1558)]

# Convert X0, X1, X2 from character to numeric
ad_data$X0 <- as.numeric(ad_data$X0)
ad_data$X1 <- as.numeric(ad_data$X1)
ad_data$X2 <- as.numeric(ad_data$X2)

# Rename columns
colnames(ad_data)[1] <- "height"
colnames(ad_data)[2] <- "width"
colnames(ad_data)[3] <- "ratio"
colnames(ad_data)[4] <- "target"

# Convert target to binary
ad_data$target <- ifelse(ad_data$target == "ad.", 1, 0)

# Display first and last 10 rows
rbind(head(ad_data, 10),
      tail(ad_data, 10))

# Count number of NA
na_counts <- colSums(is.na(ad_data))
# Percentage of NA for every col
na_percentage <- colMeans(is.na(ad_data)) * 100
na_summary <- data.frame(
  NA_Count = na_counts,
  NA_Percent = round(na_percentage, 2)
)
print(na_summary)

# Handle missing data by imputation
# Replace missing values with median
ad_data$height[is.na(ad_data$height)] <- median(ad_data$height, na.rm = TRUE)
ad_data$width[is.na(ad_data$width)] <- median(ad_data$width, na.rm = TRUE)
ad_data$ratio[is.na(ad_data$ratio)] <- median(ad_data$ratio, na.rm = TRUE)

# Recalculate Ratio
update_ratio <- function(df) {
  df$ratio[is.na(df$ratio) & !is.na(df$height) & !is.na(df$width)] <- df$height[is.na(df$ratio) & !is.na(df$height) & !is.na(df$width)] / df$width[is.na(df$ratio) & !is.na(df$height) & !is.na(df$width)]
  return(df)
}

```

```r
# Use update_ratio to replace missing values
ad_data <- update_ratio(ad_data)

# Part 2: Descriptive statistic
# Select numeric variables (excluding the binary target)
numeric_data <- ad_data[, c(1, 2, 3)]

# Calculate summary statistics with more metrics
summary_stats <- sapply(numeric_data, function(x) {
  c(Mean = mean(x),
    SD = sd(x),
    Variance = var(x),
    Min = min(x),
    Q1 = quantile(x, 0.25),
    Median = median(x),
    Q3 = quantile(x, 0.75),
    Max = max(x),
    IQR = IQR(x))
})
# Transpose and round for better readability
stats_table <- t(as.data.frame(summary_stats))
round(stats_table, 2)

# Load necessary library
library(gridExtra)

# Create a PNG file to save the table
png("C:/Users/hoang/Documents/HCMUT/242/XSTK/Assignment/datasettarget_table.png",
    width = 725, height = 150)

# Render the table
grid.table(stats_table)

# Close the PNG device to save the file
dev.off()
# Number of targets
table(ad_data$target)

# Load required libraries
library(ggplot2)

# Part 3: Visualization

# Histogram for height
ggplot(ad_data, aes(x = height)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Height", x = "Height", y = "Frequency") +
  theme_minimal()

# Histogram for width
ggplot(ad_data, aes(x = width)) +
  geom_histogram(binwidth = 10, fill = "green", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Width", x = "Width", y = "Frequency") +
  theme_minimal()

# Histogram for ratio
ggplot(ad_data, aes(x = ratio)) +
  geom_histogram(binwidth = 0.5, fill = "purple", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Ratio", x = "Ratio", y = "Frequency") +
  theme_minimal()

# Barplot for target
```

```r
ggplot(ad_data, aes(x = factor(target), fill = factor(target))) +
  geom_bar() +
  scale_fill_manual(values = c("blue", "orange")) +
  labs(title = "Barplot for Target", x = "Target (1=Ad, 0=Non-Ad)", y = "Count") +
  theme_minimal()

# Boxplot for height by target
ggplot(ad_data, aes(x = factor(target), y = height, fill = factor(target))) +
  geom_boxplot() +
  scale_fill_manual(values = c("blue", "orange")) +
  labs(title = "Boxplot of Height by Target", x = "Target", y = "Height") +
  theme_minimal()

# Boxplot for width by target
ggplot(ad_data, aes(x = factor(target), y = width, fill = factor(target))) +
  geom_boxplot() +
  scale_fill_manual(values = c("blue", "orange")) +
  labs(title = "Boxplot of Width by Target", x = "Target", y = "Width") +
  theme_minimal()

# Boxplot for ratio by target
ggplot(ad_data, aes(x = factor(target), y = ratio, fill = factor(target))) +
  geom_boxplot() +
  scale_fill_manual(values = c("blue", "orange")) +
  labs(title = "Boxplot of Ratio by Target", x = "Target", y = "Ratio") +
  theme_minimal()

# Set up 3 plots in a column layout
par(mfrow = c(3, 1), mar = c(5, 5, 4, 2))

# Strip Plot: Height vs Ad/NoAd
stripchart(height ~ target,
           data = ad_data,
           horizontal = TRUE,
           method = "jitter",
           pch = 19,
           col = rgb(0.2, 0.4, 0.8, 0.5),  # Semi-transparent blue
           main = "Height Distribution by Ad Status",
           ylab = "Ad Status (0 = Non-Ad, 1 = Ad)",
           xlab = "Height")

# Strip Plot: Width vs Ad/NoAd
stripchart(width ~ target,
           data = ad_data,
           horizontal = TRUE,
           method = "jitter",
           pch = 19,
           col = rgb(0.1, 0.7, 0.1, 0.5),  # Semi-transparent green
           main = "Width Distribution by Ad Status",
           ylab = "Ad Status (0 = Non-Ad, 1 = Ad)",
           xlab = "Width")

# Strip Plot: Ratio vs Ad/NoAd
stripchart(ratio ~ target,
           data = ad_data,
           horizontal = TRUE,
           method = "jitter",
           pch = 19,
           col = rgb(0.9, 0.2, 0.2, 0.5),  # Semi-transparent red
           main = "Aspect Ratio Distribution by Ad Status",
           ylab = "Ad Status (0 = Non-Ad, 1 = Ad)",
           xlab = "Aspect Ratio")
```

```r

# Reset layout
par(mfrow = c(1, 1))

# Select the relevant columns
pair_data <- ad_data[, c("height", "width", "ratio", "target")]

# Convert target to factor for color grouping
pair_data$target <- as.factor(pair_data$target)

library(GGally)
# Create pairwise plot
ggpairs(pair_data,
        aes(color = target, alpha = 0.6),
        upper = list(continuous = "points"),
        diag = list(continuous = "densityDiag"),
        lower = list(continuous = "smooth")) +
  theme_minimal()

# Part 4: Logistic regression model

# Logistic regression model for height
model_height <- glm(target ~ height, data = ad_data, family = "binomial")

# Draw distribution chart between 'height' and 'target'
# then draw logistic curve
plot(ad_data$height, ad_data$target,
     xlab = "Height", ylab = "Target",
     main = "Logistic Regression: Target vs Height",
     pch = 16, col = rgb(0, 0, 1, 0.3))  # Distribution chart

curve(predict(model_height, newdata = data.frame(height = x), type = "response"),
      add = TRUE, col = "red", lwd = 2)  # logistic curve

# Logistic regression model for width
model_width <- glm(target ~ width, data = ad_data, family = "binomial")

plot(ad_data$width, ad_data$target,
     xlab = "Width", ylab = "Target",
     main = "Logistic Regression: Target vs Width",
     pch = 16, col = rgb(0, 0, 1, 0.3))

curve(predict(model_width, newdata = data.frame(width = x), type = "response"),
      add = TRUE, col = "red", lwd = 2)

# Logistic regression model for ratio
model_ratio <- glm(target ~ ratio, data = ad_data, family = "binomial")

plot(ad_data$ratio, ad_data$target,
     xlab = "Ratio", ylab = "Target",
     main = "Logistic Regression: Target vs Ratio",
     pch = 16, col = rgb(0.7, 0, 0, 0.3))

curve(predict(model_ratio, newdata = data.frame(ratio = x), type = "response"),
      add = TRUE, col = "red", lwd = 2)

# Calculate the correlation matrix between the continuous variables 'height', 'width', 'ratio'
library(corrplot)
cor_matrix <- cor(ad_data[, c("height", "width", "ratio")], use = "complete.obs")

# Draw the correlation matrix plot
```

```r
corrplot(cor_matrix, method = "circle", type = "upper",
         addCoef.col = "black", tl.col = "black", number.cex = 0.8,
         title = "Correlation Matrix", mar = c(0,0,1,0))

# Part 5: Inferential Statistics
# Reset seed
set.seed(111)

# Divide the dataset into training set and testing set
# 70% - 30%
sample_index <- sample(1:nrow(ad_data), size = 0.7 * nrow(ad_data))
train_data <- ad_data[sample_index, ]
test_data <- ad_data[-sample_index, ]

# Logistic regression model with ’Height’ and ’Width’
model_hw <- glm(target ~ height + width, data = train_data, family = binomial)
summary(model_hw)

# Logistic regression model with only ’Width’
model_hw2 <- glm(target ~ width, data = train_data, family = binomial)
summary(model_hw2)

# Logistic regression model with ’Ratio’
model_ratio <- glm(target ~ ratio, data = train_data, family = binomial)
summary(model_ratio)

# Function to evaluate the model
evaluate_model <- function(model, data, label) {
  pred_prob <- predict(model, newdata = data, type = "response")
  pred_class <- ifelse(pred_prob > 0.5, 1, 0)

  actual <- data$target
  accuracy <- mean(pred_class == actual)

  cat(paste0(label, ":\n"))
  cat("Accuracy: ", round(accuracy, 4), "\n")
  cat("Confusion Matrix:\n")
  print(table(Predicted = pred_class, Actual = actual))
  cat("\n")
}

# Evaluate the model
evaluate_model(model_hw2, test_data, "Model 1 (width)")
evaluate_model(model_ratio, test_data, "Model 2 (ratio)")

# Print the AIC of models
cat("AIC Model 1 ( width): ", AIC(model_hw2), "\n")
cat("AIC Model 2 (ratio): ", AIC(model_ratio), "\n")

  library(pROC)
  # Prediction on the test dataset
  prob_hw <- predict(model_hw2, newdata = test_data, type = "response")
  prob_ratio <- predict(model_ratio, newdata = test_data, type = "response")

  # Plot ROC
  roc_hw <- roc(test_data$target, prob_hw)
  roc_ratio <- roc(test_data$target, prob_ratio)

  # Plot comparison chart
  plot(roc_hw, col = "blue", lwd = 2, main = "ROC Curve Comparison")
  lines(roc_ratio, col = "red", lwd = 2)
  legend("bottomright", legend = c("Model 1: width", "Model 2: ratio"),
         col = c("blue", "red"), lwd = 2)

  # Print AUC
  cat("AUC Model 1 (width): ", auc(roc_hw), "\n")
  cat("AUC Model 2 (ratio): ", auc(roc_ratio), "\n")
```