Group 23

```
library(tidyverse)
library(janitor)
library(ggplot2)
library(moderndive)
library(gapminder)
library(sjPlot)
library(stats)
library(jtools)
library(GGally)
library(gt)
library(pROC)
library(randomForest)
library(caret)
```

1 Introduction

The surge in spam emails leads to wasted time, security risks, and financial losses. To address this issue, companies rely on spam filters to automatically classify emails and minimize disruptions. Therefore, studying the key text features that influence whether an email is classified as spam is crucial. **Research Question** Which text characteristics influence whether an email will be classified as spam or not?

2 Exploratory Data Analysis

2.1 Description of Data

2.1.1 Data Source

This dataset were collected at Hewlett-Packard Labs and shared with the UCI Machine Learning Repository. The dataset contains data on classifying emails as spam or not spam, with additional variables indicating the frequency of certain words and characters in the email.

```
data<-read.csv('~/Desktop/Group_23/dataset23.csv')
data$yesno<-as.factor(data$yesno)
data <- data[rowSums(data[, 2:6] > 1) == 0, ]
# the percentage of total numbe can not be greater than 1
```

2.1.2 Varaible Description

This table provides variable descriptions to ensure clarity in their interpretation.

```
variable <- data.frame(
  VariableName = colnames(data),
  Description = c("Total length of uninterrupted sequences of capitals ",
                  "Occurrences of the dollar sign,
                  as a percentage of total number of characters ",
                  " Occurrences of ‘!’,
                  as a percentage of total number of characters ",
                  "Occurrences of ‘money’,
                  as a percentage of total number of characters",
                  "Occurrences of the string ‘000’,
                  as a percentage of total number of characters ",
                  "Occurrences of ‘make’,
                  as a percentage of total number of characters ",
                  "A factor variable indicating if the email was spam,
                  ‘y’, or not spam, ‘n’"),
  Type = sapply(data, class)
)
variable %>%
  gt() %>%
  tab_header(title = "Description of Variables")%>%
  tab_options(
```

Table 1: Description of Variables

Description of Variables

VariableName	Description	Type
crl.tot	Total length of uninterrupted sequences of capitals	integer
dollar	Occurrences of the dollar sign, as a percentage of total number of characters	numeric
bang	Occurrences of '!', as a percentage of total number of characters	numeric
money	Occurrences of 'money', as a percentage of total number of characters	numeric
n000	Occurrences of the string '000', as a percentage of total number of characters	numeric
make	Occurrences of 'make', as a percentage of total number of characters	numeric
yesno	A factor variable indicating if the email was spam, 'y', or not spam, 'n'	factor

```
    table.font.size = "small",
    table.width = pct(115))
```

2.1.3 Data Visualization

2.1.3.1 summary of data

This table shows the summary of the data.

```
data |>
  summarize(
    crl.tot = mean(crl.tot),
    dollar = mean(dollar),
    bang = mean(bang),
    money = mean(money),
    n000 = mean(n000),
    make = mean(make),
             .by = yesno) |>
  gt() |>
  tab_header(
 title= "mean of Variables") |>
  fmt_number(decimals=2)
```

Table 2 shows that for the variable(crl.tot): The average value in spam emails (409.26) is significantly higher than in non-spam emails (157.83). For the variable (dollar): the occurrence of the "$" symbol in spam emails is 13%, whereas in non-spam emails, it is only 1%. For the variable (bang!): the "!" symbol appears in 31% of spam emails, while in non-spam emails, it

Table 2: summary of mean

mean of Variables

yesno	crl.tot	dollar	bang	money	n000	make
n	157.83	0.01	0.05	0.01	0.00	0.03
y	409.26	0.13	0.31	0.15	0.15	0.11

Table 3: summary of median

median of Variables

yesno	crl.tot	dollar	bang	money	n000	make
n	54.00	0.00	0.00	0.00	0.00	0.00
y	190.50	0.06	0.25	0.00	0.00	0.00

appears only 5% of the time. For the word "money" occurrence: The word "money" appears in 15% of spam emails, whereas in non-spam emails, it appears only 1% of the time. For the occurrence of "000" (n000): "000" appears in 15% of spam emails, but is almost nonexistent (0%) in non-spam emails. For the word "make" occurrence: The word "make" appears in 11% of spam emails, while in non-spam emails, it appears only 3% of the time.

```
data |>
  summarize(
    crl.tot = median(crl.tot),
    dollar = median(dollar),
    bang = median(bang),
    money = median(money),
    n000 = median(n000),
    make = median(make),
    .by = yesno) |>
  gt() |>
  tab_header(
 title= "median of Variables") |>
  fmt_number(decimals=2)
```

Table 3 shows that for the variable (crl.tot): The median value in spam emails (190.50) is significantly higher than in non-spam emails (54.00). For the variable (dollar): The occurrence of the "$" symbol in spam emails has a median value of 0.06, whereas in non-spam emails,it is 0.00. For the variable (bang!): The "!" symbol appears in 25% of spam emails, while in non-spam emails, the median value is 0.00. For the word "money" occurrence: The median

occurrence of "money" is 0.00 for both spam and non-spam emails. For the occurrence of "000" (n000): The median occurrence of "000" is 0.00 for both spam and non-spam emails. For the word "make" occurrence: The median occurrence of "make" is 0.00 for both spam and non-spam emails.

Based on the @table1 and @table2,the result shows that most mean values greater than median values may indicate right skewness.

2.1.3.2 correlation

the followed figure shows the correlation between the variables.

```
cor_matrix <- cor(data[, c("crl.tot", "dollar", "bang", "money", "n000", "make")])
corrplot::corrplot(cor_matrix, method = "number")
```

Figure 1: correlation between variables

As Figure 1 shows, The maximum correlation between variables is 0.4, indicating that the correlation between variables is not high. Therefore, interaction terms will not be considered in the model.

2.1.3.3 Variable Visualization

```
ggplot(data, aes(x=yesno, y=crl.tot, fill=yesno)) +
  geom_boxplot()
```

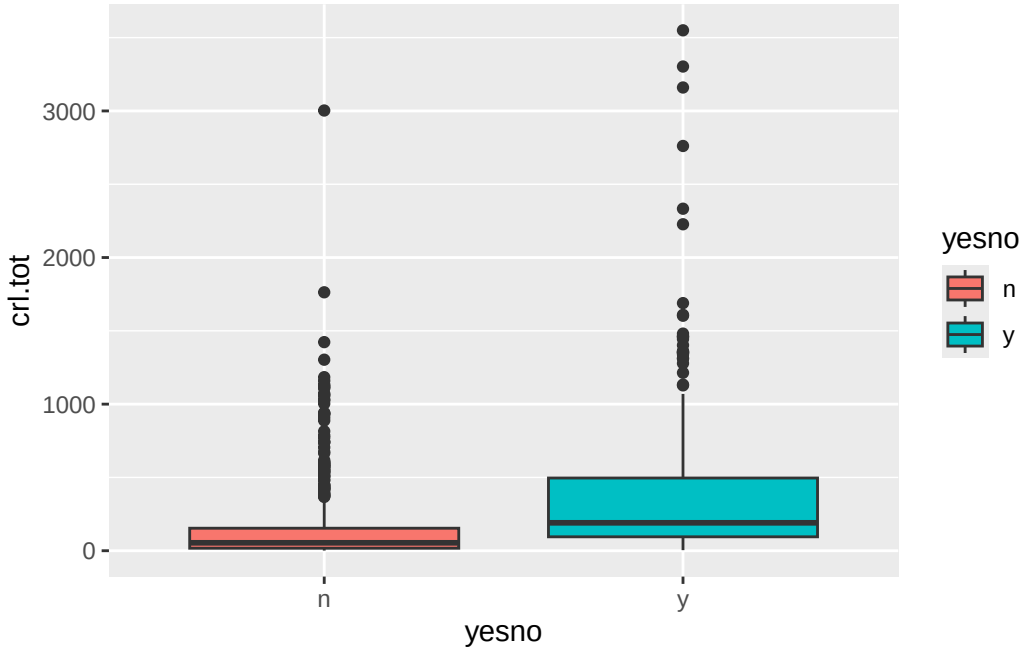

Figure 2: Distribution of crl.tot between spam and non-spamemails

Figure 2 compares the distribution of the variable crl.tot between spam (y) and non-spam (n) emails.The median crl.tot value is higher in spam emails (y).Spam emails (y) have a larger interquartile range (IQR), indicating greater variability in crl.tot.Spam emails contain more extreme outliers, with some emails having crl.tot values exceeding 3000.Non-spam emails (n) have lower crl.tot values and a more concentrated distribution, with a lower median and fewer outliers compared to spam emails.

```
ggplot(data, aes(x=crl.tot, fill=yesno)) +
  geom_density(alpha=0.5)
```

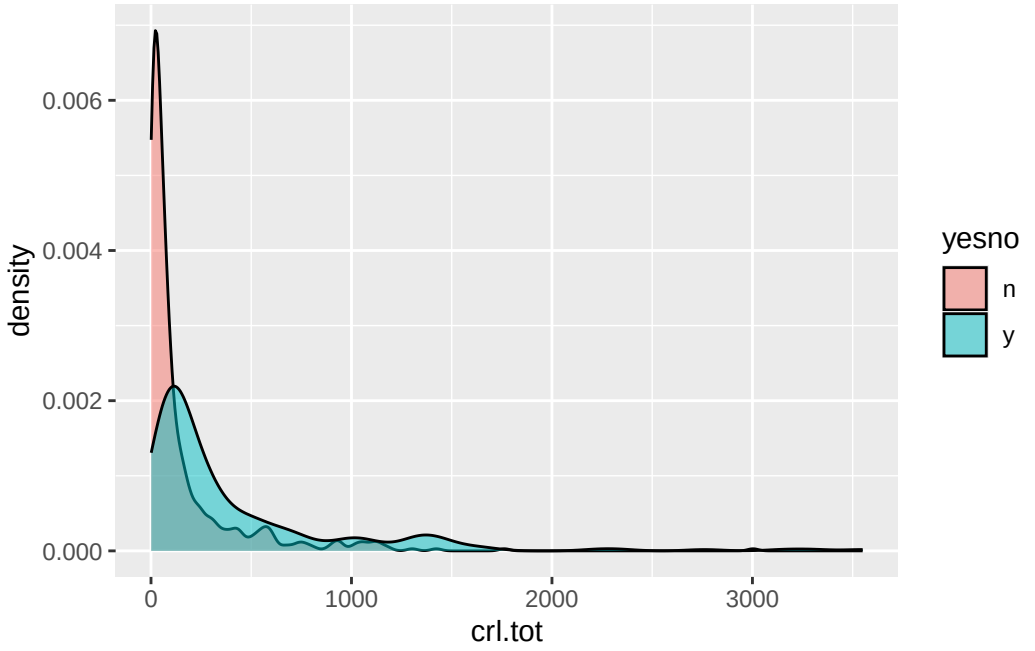

Figure 3: Distribution of crl.tot in Spam (y) and Non-Spam (n) Emails

Figure 3 visualizes the distribution of the variable crl.tot for spam (y) and non-spam (n) emails. Non-spam emails (n) are highly concentrated around low crl.tot values. The pink region represents non-spam emails, forming a sharp peak near 0, indicating that most non-spam emails have very short uninterrupted capital letter sequences. However, spam emails (y) have a broader and more dispersed distribution. The blue region represents spam emails, which exhibit a right-skewed distribution. The distribution of spam emails shows a long tail, where although most spam emails have relatively small crl.tot values, some emails have extremely high crl.tot values, even exceeding 3000.

```
ggplot(data, aes(x=yesno, y=bang, fill=yesno)) +
  geom_boxplot()
```

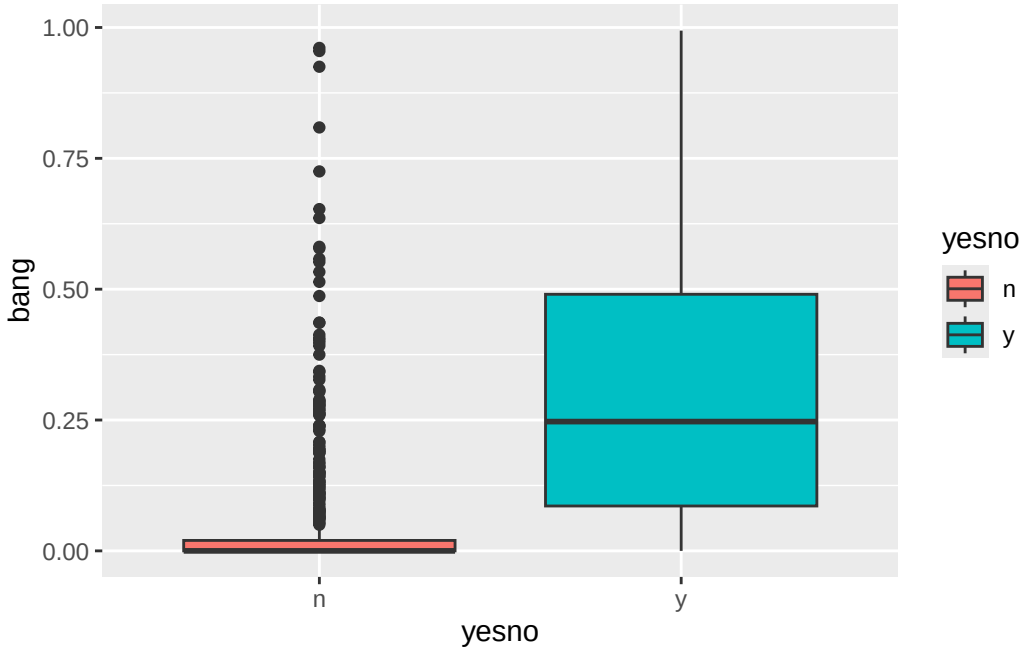

Figure 4: Distribution of bang in Spam (y) and Non-Spam (n) Emails

Figure 4 shows the the distribution of bang for spam (y) and non-spam (n) emails. Bang values in non-spam emails (n) are very low. The pink box (non-spam emails) is concentrated near 0, indicating that most non-spam emails contain almost no exclamation marks. There are several outliers. However, spam emails (y) use significantly more exclamation marks. The blue box (spam emails) is wider, and the median bang value is much higher than that of non-spam emails. The interquartile range (IQR) is larger, indicating a greater variation in the use of exclamation marks in spam emails. The maximum bang value in spam emails is much higher than in non-spam emails.

```
ggplot(data, aes(x=bang, fill=yesno)) +
  geom_density(alpha=0.5)
```

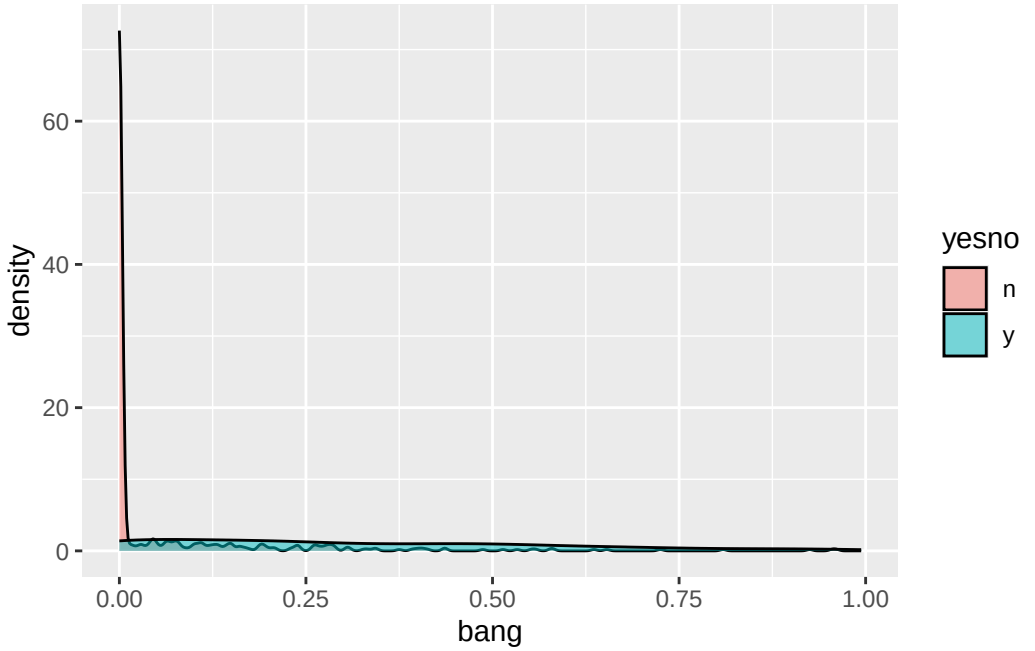

Figure 5: Distribution of bang in Spam (y) and Non-Spam (n) Emails

Figure 5 visualizes the distribution of bang for spam (y) and non-spam (n) emails. Non-spam emails (n) are highly concentrated at 0. The pink density area forms an extreme peak at bang = 0, indicating that most non-spam emails contain almost no exclamation marks (!).However, spam emails (y) have a more dispersed distribution.

```
ggplot(data, aes(x=yesno, y=money, fill=yesno)) +
  geom_boxplot()
```

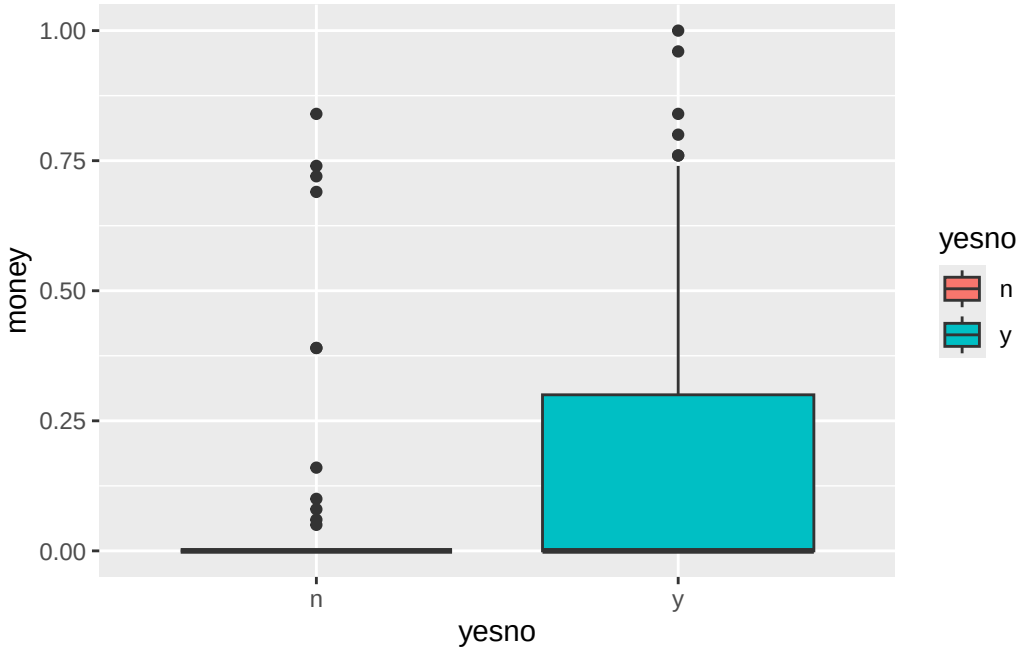

Figure 6: Distribution of money in Spam (y) and Non-Spam (n) Emails

Figure 6 visualizes the distribution of money for spam (y) and non-spam (n) emails. Non-spam emails (n) rarely contain the word "money". The pink box (non-spam emails) is fully concentrated around 0, but there are a few outliers. However, spam emails (y) have a significantly higher proportion of "money". The blue box (spam emails) is much larger than that of non-spam emails, with a higher median and interquartile range (IQR), and it also contains many extreme outliers.

```
ggplot(data, aes(x=money, fill=yesno)) +
  geom_density(alpha=0.5)
```

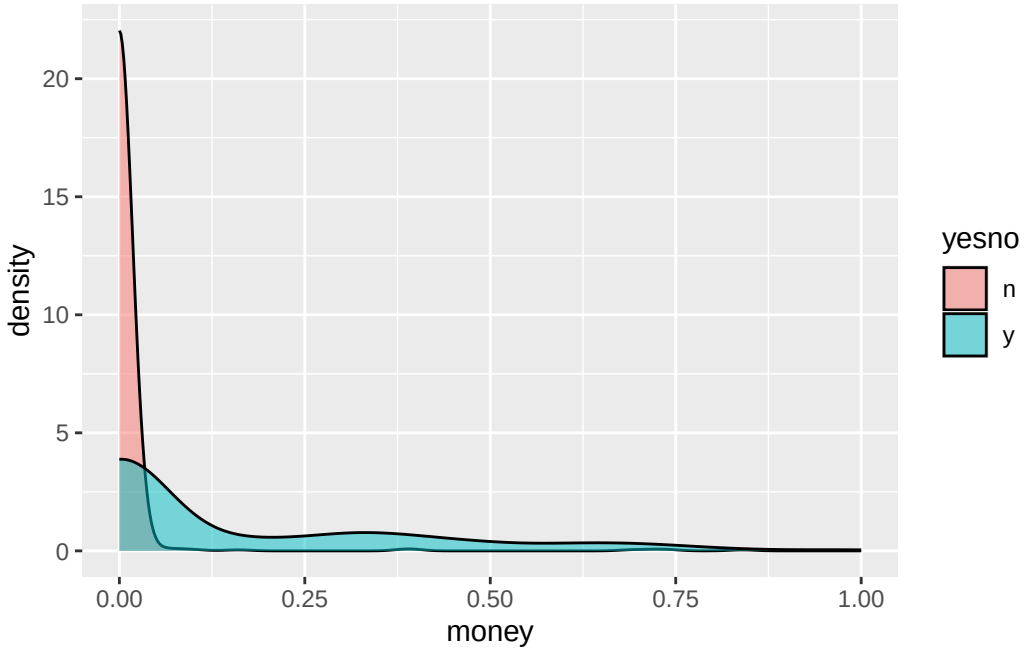

Figure 7: Distribution of money in Spam (y) and Non-Spam (n) Emails

Figure 7 visualizes the distribution of money for spam (y) and non-spam (n) emails.Non-spam emails (n) are highly concentrated at 0. The pink density area forms an extreme peak at 0.However, spam emails (y) have a broader distribution. The blue density area is more dispersed.

```
ggplot(data, aes(x=yesno, y=dollar, fill=yesno)) +
  geom_boxplot()
```

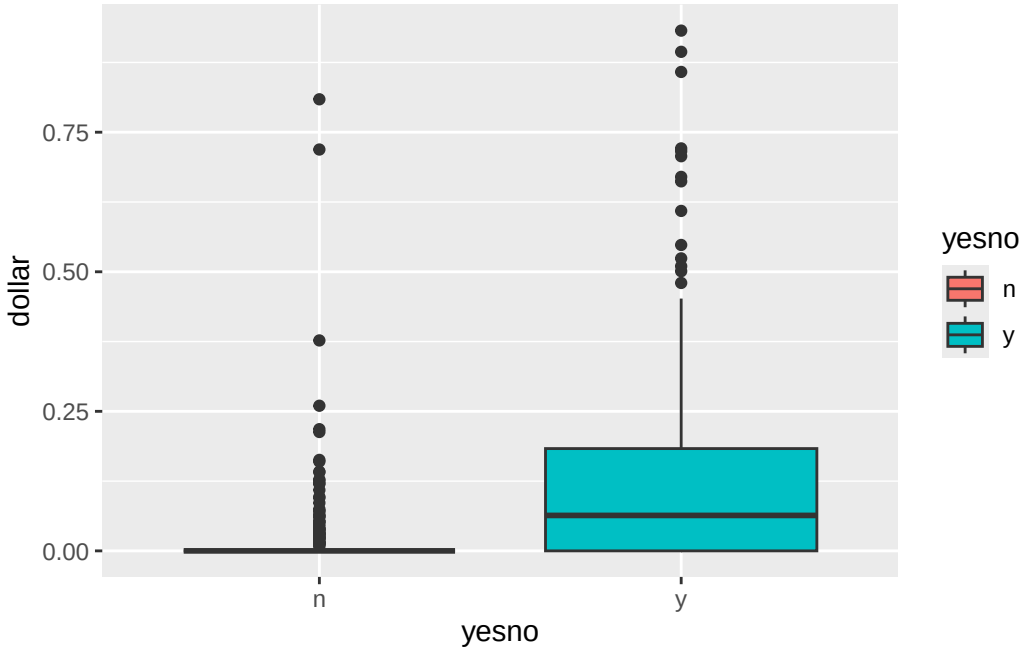

Figure 8: Distribution of dollar in Spam (y) and Non-Spam (n) Emails

Figure 8 shows the distribution of dollar for spam (y) and non-spam (n) emails. Non-spam emails (n) rarely contain the dollar sign. The pink box (non-spam emails) is fully concentrated at 0, with a few outliers. In contrast, spam emails (y) have a significantly higher proportion of dollar signs. The blue box (spam emails) is much larger than that of non-spam emails, with a higher median and interquartile range (IQR). Additionally, there are many extreme outliers.

```
ggplot(data, aes(x=dollar, fill=yesno)) +
  geom_density(alpha=0.5)
```

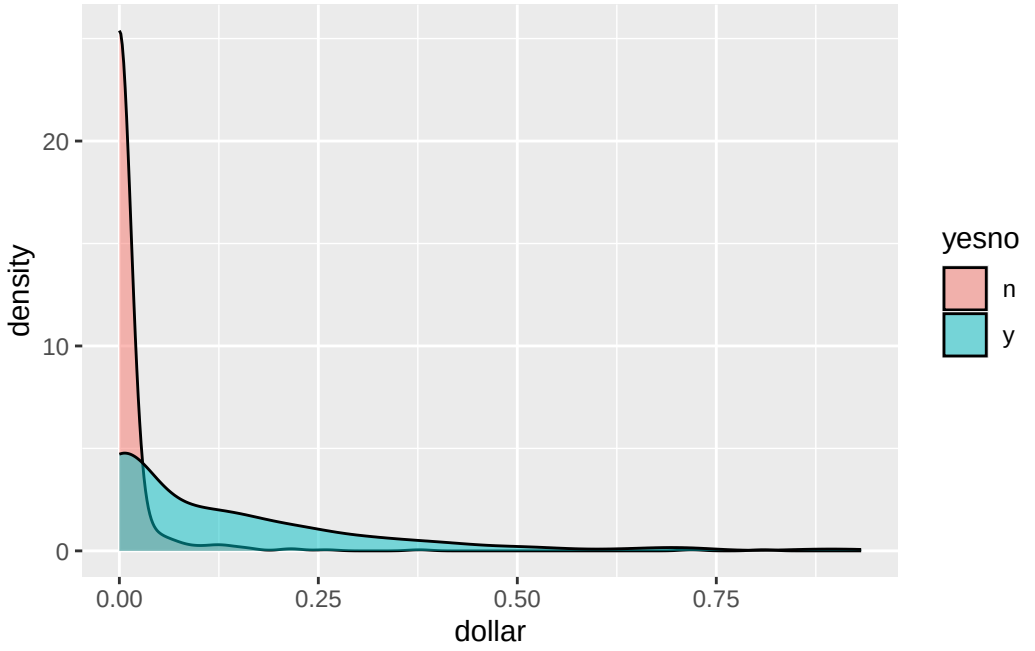

Figure 9: Distribution of dollar in Spam (y) and Non-Spam (n) Emails

Figure 9 shows that the distribution of dollar for spam (y) and non-spam (n) emails. Non-spam emails (n) are highly concentrated at dollar = 0. The pink density area forms an extreme peak at dollar = 0. However, spam emails (y) have a broader distribution. The blue density area is more dispersed.

```
ggplot(data, aes(x=yesno, y=n000, fill=yesno)) +
  geom_boxplot()
```

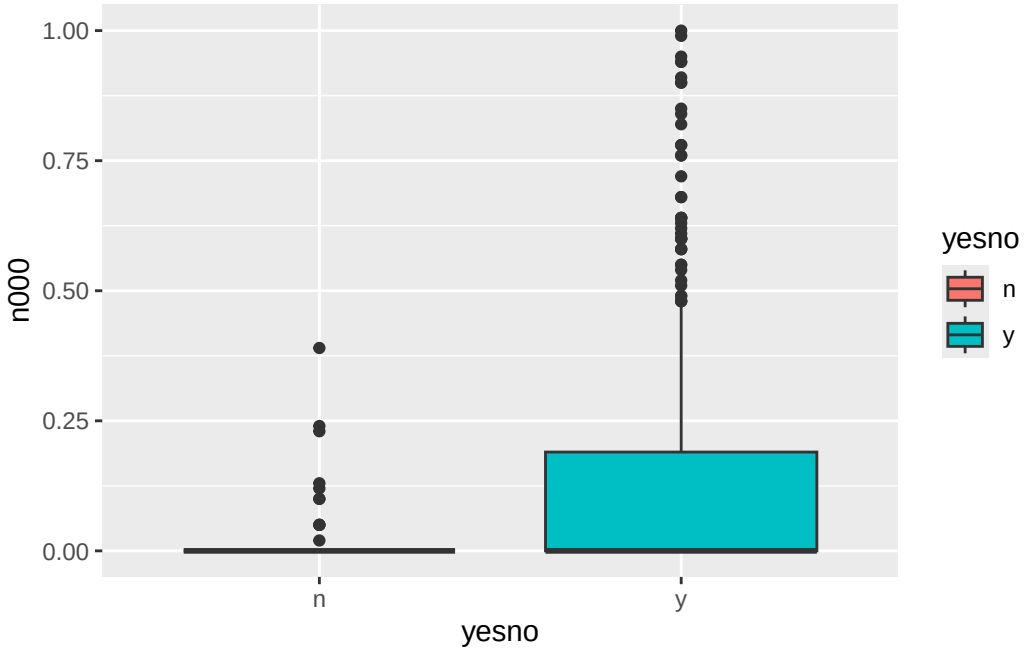

Figure 10: Distribution of n000 for spam (y) and non-spam (n) emails

Figure 10 shows the distribution of n000 for spam (y) and non-spam (n) emails.Non-spam emails (n) rarely contain "000". The pink box (non-spam emails) is fully concentrated at 0, with a few outliers. However, the blue box (spam emails) is much larger than that of non-spam emails, with a higher median and interquartile range (IQR). The distribution of spam emails is more spread out and contains many extreme outliers.

```
ggplot(data, aes(x=n000, fill=yesno)) +
  geom_density(alpha=0.5)
```

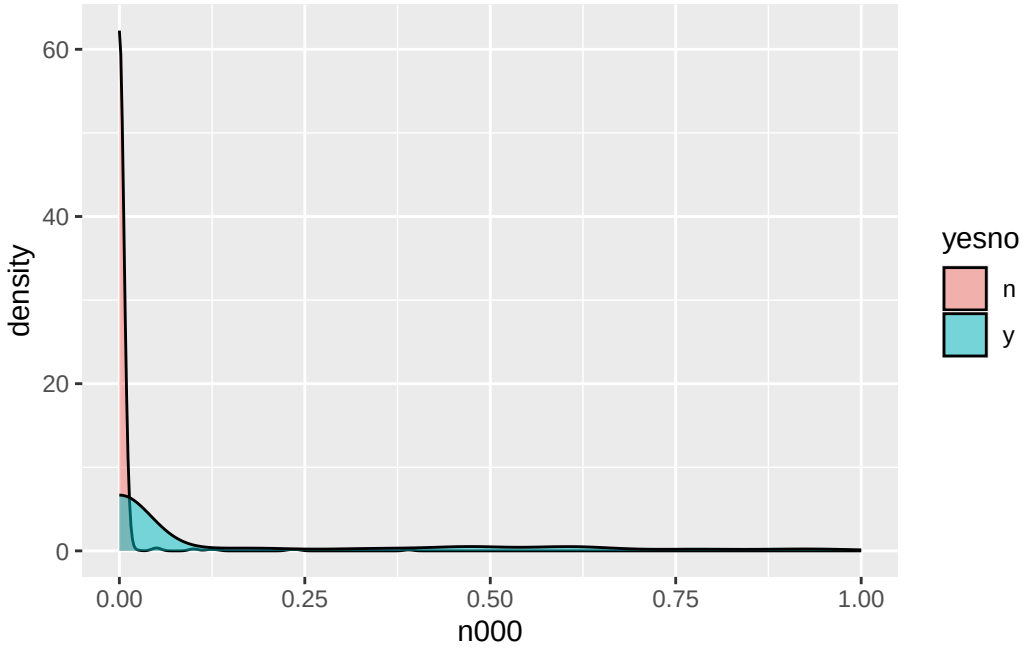

Figure 11: Distribution of n000 in Spam (y) and Non-Spam (n) Emails

Figure 11 shows the distribution of n000 for spam (y) and non-spam (n) emails. Non-spam emails (n) are highly concentrated at 0, with the pink density area forming an extreme peak at 0. Spam emails (y) have a broader distribution.

```
ggplot(data, aes(x=yesno, y=make, fill=yesno)) +
  geom_boxplot()
```

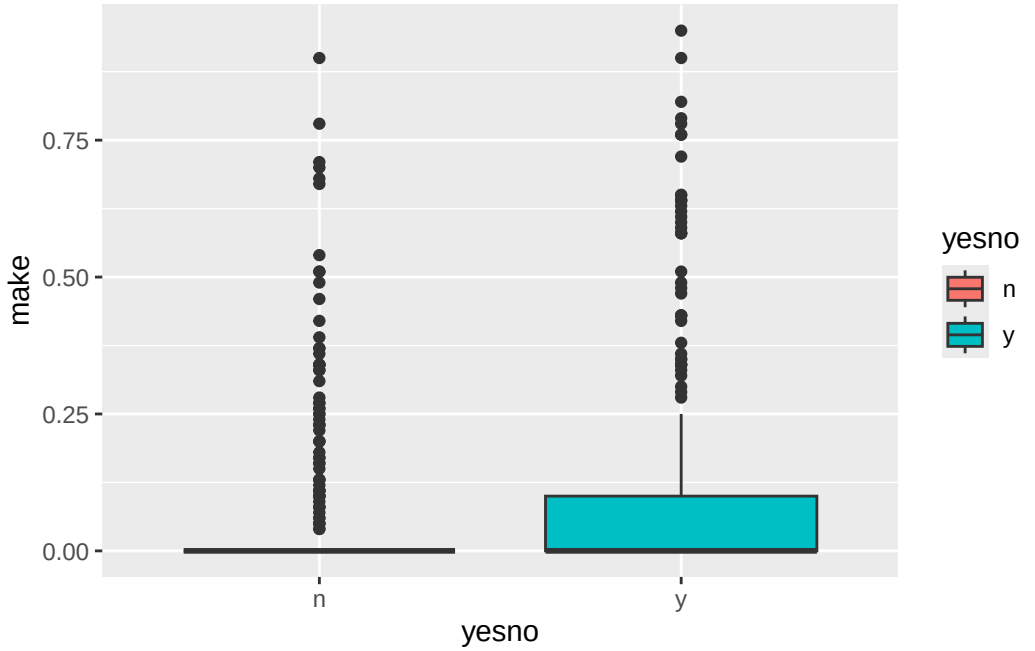

Figure 12: Distribution of make for spam (y) and non-spam (n) emails

Figure 12 the distribution of make for spam (y) and non-spam (n) emails.Non-spam emails (n) rarely contain "make". The pink box (non-spam emails) is fully concentrated at 0, with multiple outliers. The proportion of "make" is significantly higher in spam emails (y). The blue box (spam emails) is larger than that of non-spam emails, with a higher median and interquartile range (IQR). The distribution of spam emails is more spread out and contains many extreme outliers.

```
ggplot(data, aes(x=make, fill=yesno)) +
  geom_density(alpha=0.5)
```

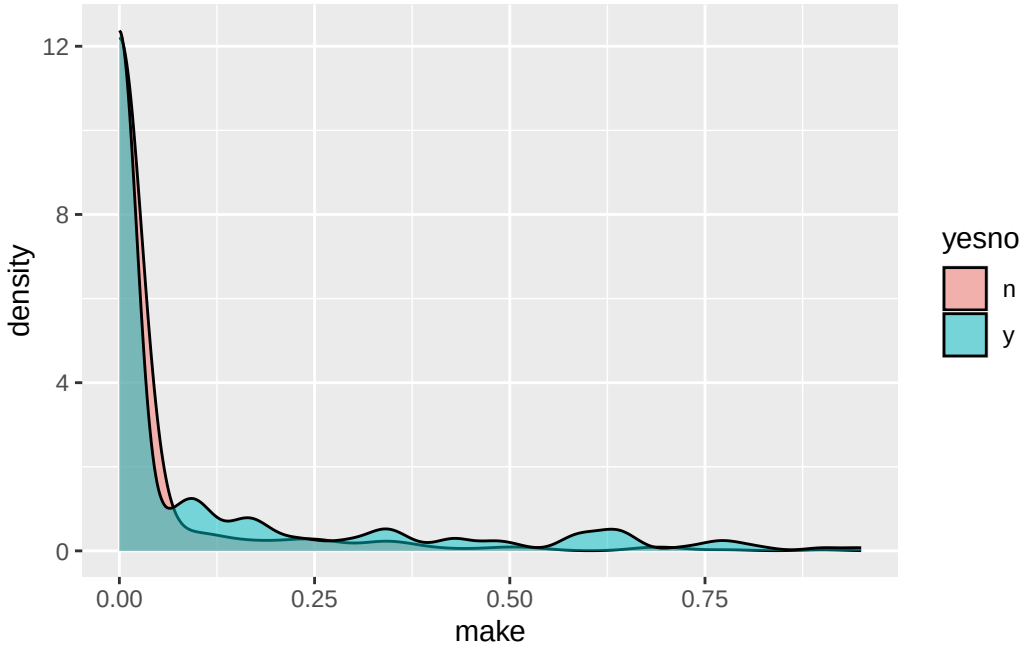

Figure 13: Distribution of make in Spam (y) and Non-Spam (n) Emails

Figure 13 shows the distribution of make (the proportion of the word "make" in the email) for spam (y) and non-spam (n) emails.Non-spam emails (n) are highly concentrated at 0, with the pink density area forming an extreme peak at 0. Spam emails (y) have a broader distribution, with the blue density area being more dispersed.

2.2 Data Preprocessing

Replace values below the 1st percentile with the 1st percentile value and values over the 99th percentile with the 99th percentile value, then standardize the data to mitigate the effects of outliers and right skewness. Due to the wide distribution and right-skewed nature of crl.tot, we substitute it with log(data$crl.tot + 1).

```
data1<-data
win <- function(x, lower_perc = 0.01, upper_perc = 0.99) {
  x <- as.numeric(x)
  q <- quantile(x, probs = c(lower_perc, upper_perc), na.rm = TRUE)
  x[x < q[1]] <- q[1]
  x[x > q[2]] <- q[2]
  return(x)
}
```

```r
numeric_vars <- c("crl.tot", "dollar", "bang", "money", "n000", "make")
data[numeric_vars] <- lapply(data[numeric_vars], win)
data[,1:6]<-scale(data[,1:6])
data$crl.tot_log <- log(data$crl.tot+1)
```

3 Modelling

In the modeling section, data preprocessing was first conducted, including winsorization (outlier removal), standardization, and logarithmic transformation. Subsequently, three logistic regression models were constructed: one using the original data, one with standardized variables, and another incorporating log-transformed variables. The Akaike Information Criterion (AIC) was employed to compare model performance. Finally, regression coefficients and predictive outcomes were visualized. The ultimate objective was to identify the optimal transformation of predictor variables to enhance model performance.

$$Y_i \sim \text{Bernoulli}(p_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 \text{crl.tot}_i + \beta_2 \text{dollar}_i + \beta_3 \text{bang}_i + \beta_4 \text{money}_i + \beta_5 \text{n000}_i + \beta_6 \text{make}_i$$

3.1 Variable Definitions

- Y_i: A binary factor variable indicating if the email was spam (`y`) or not (`n`).
- crl.tot_i: Total length of uninterrupted sequences of capital letters.
- dollar_i: Occurrences of the dollar sign (`$`), as a percentage of total number of characters.
- bang_i: Occurrences of `!`, as a percentage of total number of characters.
- money_i: Occurrences of the word "money", as a percentage of total number of characters.
- n000_i: Occurrences of the string "000", as a percentage of total number of characters.
- make_i: Occurrences of the word "make", as a percentage of total number of characters.

3.2 Alternative Model with Log Transformation

An alternative model is estimated with a log transformation of crl.tot:

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 log(\text{crl.tot}_i) + \beta_2 \text{dollar}_i + \beta_3 \text{bang}_i + \beta_4 \text{money}_i + \beta_5 \text{n000}_i + \beta_6 \text{make}_i$$

- Y_i: A binary factor variable indicating if the email was spam (`y`) or not (`n`).
- $log(\text{crl.tot}_i)$: The logarithm-transformed length of uninterrupted sequences of capital letters.
- dollar_i: Occurrences of the dollar sign (`$`), as a percentage of total number of characters.
- bang_i: Occurrences of `!`, as a percentage of total number of characters.
- money_i: Occurrences of the word "money", as a percentage of total number of characters.
- n000_i: Occurrences of the string "000", as a percentage of total number of characters.
- make_i: Occurrences of the word "make", as a percentage of total number of characters.

3.3 Logistic regression models

```
model_original <- glm(yesno ~ crl.tot + dollar + bang + money + n000 + make,
             family = binomial(link = "logit"),
             data = data1)

summary(model_original)
```

```
Call:
glm(formula = yesno ~ crl.tot + dollar + bang + money + n000 +
    make, family = binomial(link = "logit"), data = data1)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.3209859  0.1563256 -14.847  < 2e-16 ***
crl.tot      0.0007367  0.0002964   2.486 0.012924 *
dollar       7.1037071  1.5468059   4.593 4.38e-06 ***
bang         5.0489232  0.5471349   9.228  < 2e-16 ***
money        4.8483493  1.0004289   4.846 1.26e-06 ***
n000         7.6756922  2.0675084   3.713 0.000205 ***
make        -0.7336087  0.9511158  -0.771 0.440521
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1008.14  on 798  degrees of freedom
Residual deviance:  578.74  on 792  degrees of freedom
AIC: 592.74

Number of Fisher Scoring iterations: 7
```

```r
model_scale <- glm(yesno ~ crl.tot + dollar + bang + money + n000 + make,
             family = binomial(link = "logit"),
             data = data)

summary(model_scale)
```

```
Call:
glm(formula = yesno ~ crl.tot + dollar + bang + money + n000 +
    make, family = binomial(link = "logit"), data = data)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.5182     0.1385  -3.742 0.000183 ***
crl.tot       0.2906     0.1217   2.388 0.016934 *
dollar        0.8696     0.1824   4.767 1.87e-06 ***
bang          1.1251     0.1216   9.250  < 2e-16 ***
money         0.7472     0.1542   4.844 1.27e-06 ***
n000          1.2408     0.3352   3.702 0.000214 ***
make         -0.1137     0.1449  -0.785 0.432501
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1008.14  on 798  degrees of freedom
Residual deviance:  575.17  on 792  degrees of freedom
AIC: 589.17

Number of Fisher Scoring iterations: 7
```

```r
model_scale_log <- glm(yesno ~ bang + crl.tot_log + dollar+money+n000+make,
               family = binomial(link = "logit"), data = data)
summary(model_scale_log)
```

```
Call:
glm(formula = yesno ~ bang + crl.tot_log + dollar + money + n000 +
    make, family = binomial(link = "logit"), data = data)

Coefficients:
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.3773     0.1367  -2.761 0.005767 **
bang          1.1332     0.1222   9.277  < 2e-16 ***
crl.tot_log   0.6552     0.1645   3.984 6.78e-05 ***
dollar        0.8414     0.1776   4.737 2.17e-06 ***
money         0.7068     0.1528   4.627 3.71e-06 ***
n000          1.1532     0.3164   3.644 0.000268 ***
make         -0.1511     0.1473  -1.026 0.304915
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1008.14  on 798  degrees of freedom
Residual deviance:  565.11  on 792  degrees of freedom
AIC: 579.11

Number of Fisher Scoring iterations: 7
```

```
AIC(model_original,model_scale,model_scale_log)
```

```
                df      AIC
model_original   7 592.7427
model_scale      7 589.1703
model_scale_log  7 579.1087
```

```
BIC(model_original,model_scale,model_scale_log)
```

```
                df      BIC
model_original   7 625.5262
model_scale      7 621.9538
model_scale_log  7 611.8922
```

The Akaike Information Criterion (AIC) values indicate the performance of the three logistic regression models: - model_original: AIC = 592.74 - model_scale: AIC = 589.17 (improved after standardization) - model_scale_log: AIC = 579.11 (best-performing model with log transformation)

Since a lower AIC value indicates a better fit, the log-transformed model (`model_scale_log`) is the optimal choice, as it achieves the lowest AIC.

```
plot_model(model_scale_log, show.values = TRUE, show.p = TRUE)
```

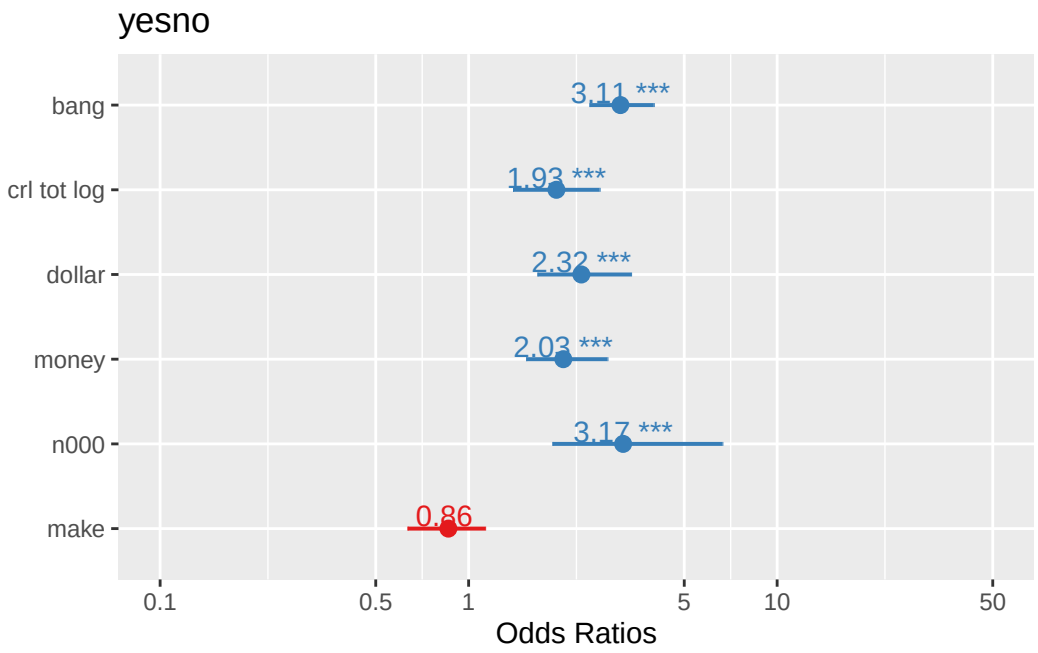

Figure 14: Coefficient Estimates with p-Values

Figure 14 displays the estimated odds ratios: Significant Positive Effects: `bang` (3.11), `crl.tot_log` (1.93), `dollar` (2.32), `money` (2.03), and `n000` (3.17) all have positive effects on the probability of an email being spam. This means that higher occurrences of these features increase the likelihood of spam classification. Non-Significant / Negative Effect: `make` (0.86) has a negative effect and is not statistically significant, meaning it does not contribute to predicting spam effectively.

```
plot_model(model_scale_log, type = "pred", title = "",col='steelblue')
```

```
$bang
```

```
$crl.tot_log
```

```
$dollar
```

```
$money

$n000

$make
```

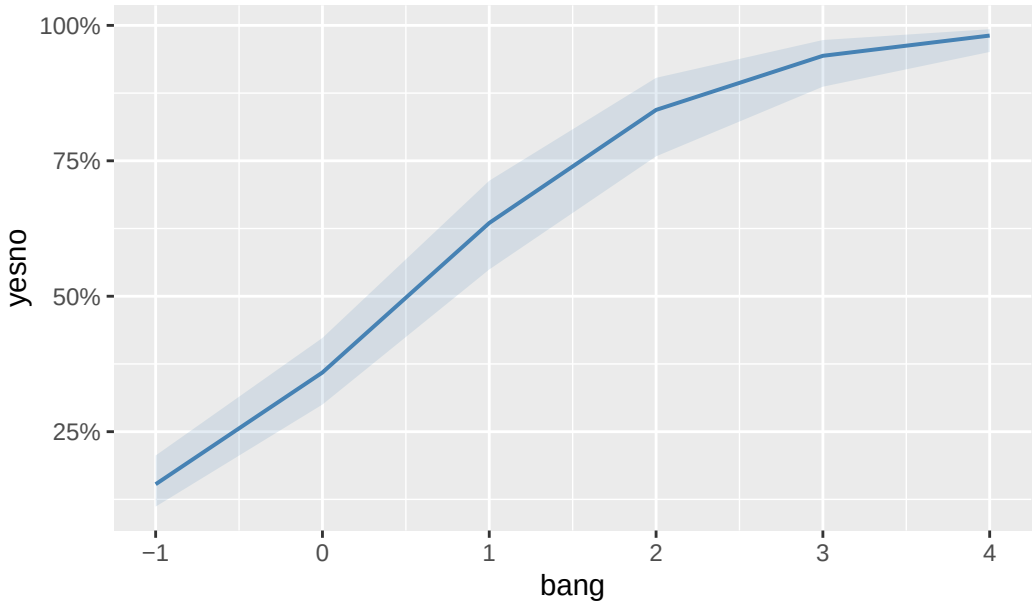

Figure 15: Marginal Predicted Effects

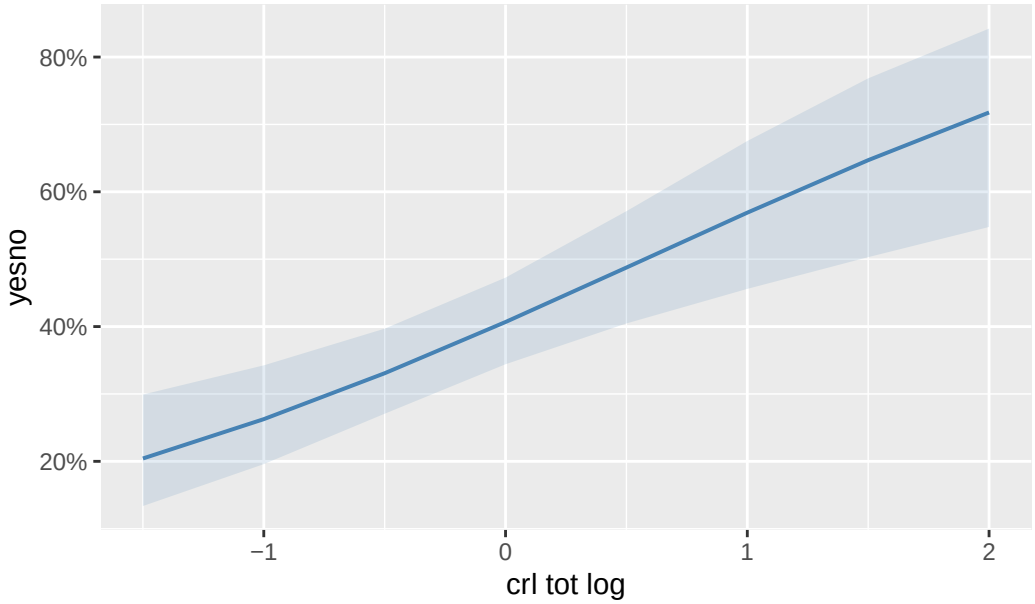

Figure 16: Marginal Predicted Effects

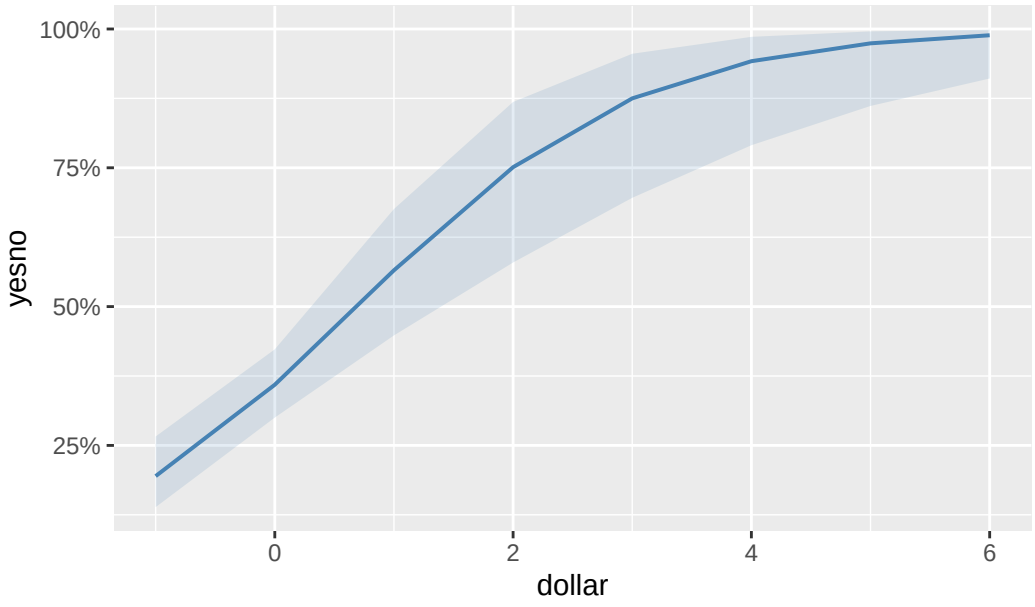

Figure 17: Marginal Predicted Effects

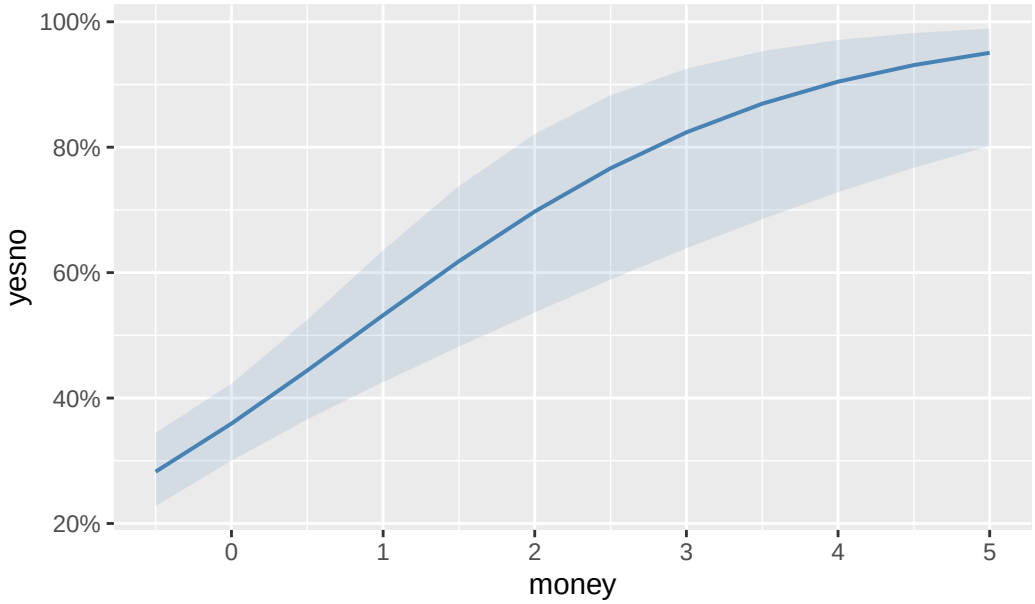

Figure 18: Marginal Predicted Effects

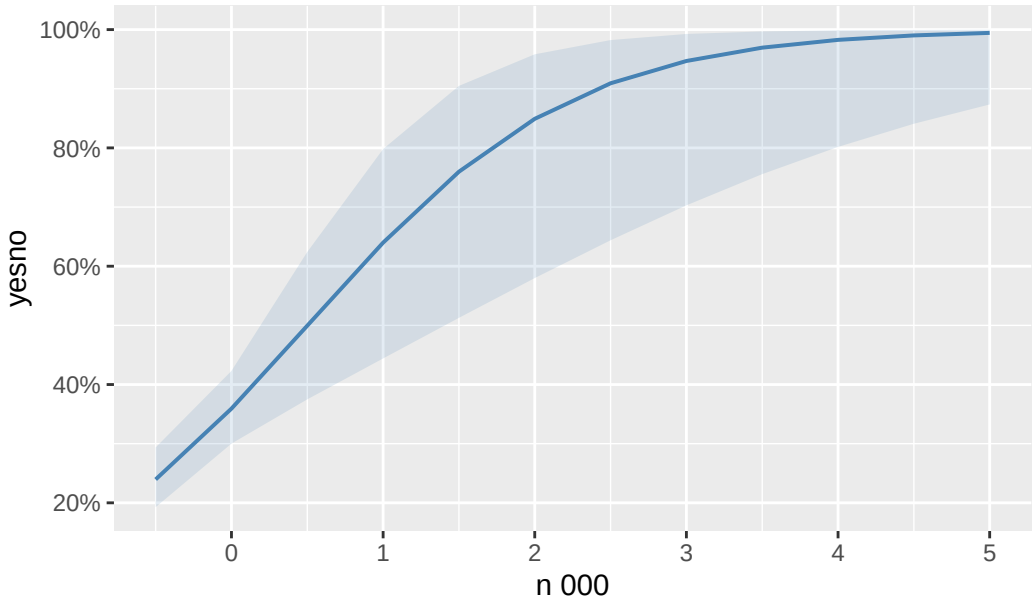

Figure 19: Marginal Predicted Effects

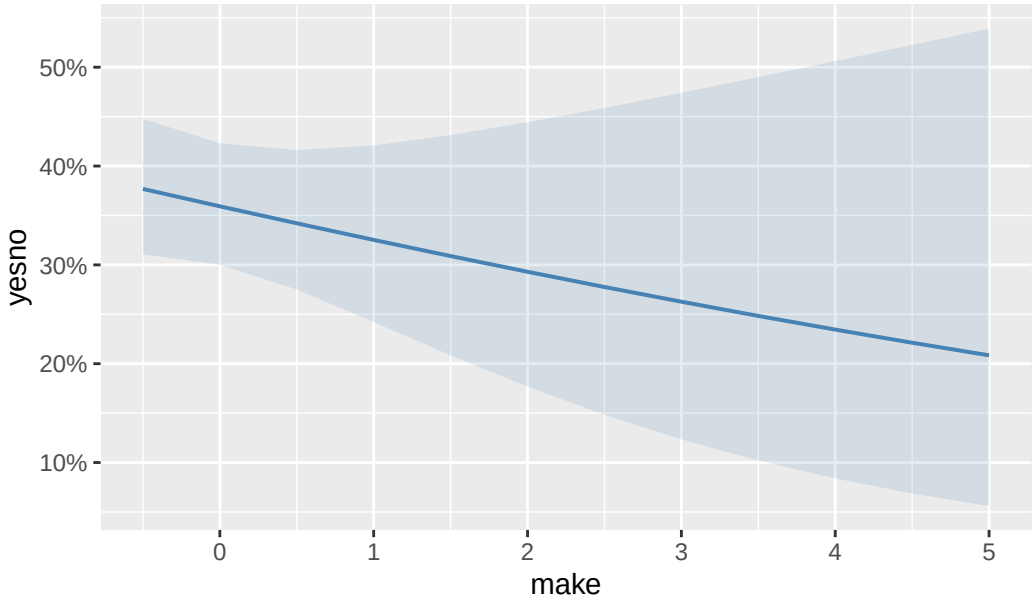

Figure 20: Marginal Predicted Effects

The plots display the estimated odds ratios: Significant Positive Effects: `bang` (3.11), `crl.tot_log` (1.93), `dollar` (2.32), `money` (2.03), and `n000` (3.17) all have positive effects on the probability of an email being spam. This means that higher occurrences of these features increase the likelihood of spam classification. Non-Significant / Negative Effect: `make` (0.86) has a negative effect and is not statistically significant, meaning it does not contribute to predicting spam effectively.

4 Conclusion

The analysis demonstrates that log transformation of `crl.tot` improves model performance, as indicated by the lowest AIC value in `model_scale_log`. Key predictors of spam classification include `bang`, `dollar`, `money`, `n000`, and `crl.tot_log`, all of which significantly increase the likelihood of an email being spam. In contrast, `make` has a negligible or even negative effect, suggesting it may not be a useful predictor.

5 Further Work

In the further work, we use the Random Forest model to explore which text characteristics influence whether an email is classified as spam, and further evaluate the GLM model.

```
set.seed(123)
index <- createDataPartition(data$yesno, p = 0.7, list = FALSE)
train_data <- data[index, ]
test_data <- data[-index, ]
```

In this step, we split the dataset, allocating 70% of the data for training the model (train_data) and the remaining 30% for testing the model (test_data), ensuring its performance on new data.

```
glm_model <- glm(yesno ~ bang + crl.tot_log + dollar+money+n000+make, data = train_data, fami

glm_pred_prob <- predict(glm_model, newdata = test_data, type = "response")
glm_pred_class <- ifelse(glm_pred_prob > 0.5,'y','n')
glm_confusion <- confusionMatrix(factor(glm_pred_class), factor(test_data$yesno))
glm_roc <- roc(test_data$yesno, glm_pred_prob)
```

In this step, we use GLM to train a logistic regression model, which is applied to a binary classification task. Once the training is complete, we make predictions on the test set and then evaluate its classification performance using a confusion matrix and an ROC curve.

```
set.seed(123)
rf_model <- randomForest(yesno ~ bang + crl.tot_log + dollar+money+n000+make, data = train_da

rf_pred_prob <- predict(rf_model, newdata = test_data, type = "prob")[, 2]
rf_pred_class <- ifelse(rf_pred_prob > 0.5,'y','n')

rf_confusion <- confusionMatrix(factor(rf_pred_class), factor(test_data$yesno))
rf_roc <- roc(test_data$yesno, rf_pred_prob)
```

In this step, we use the Random Forest model to train a spam classifier and make predictions on the test set (test_data). The model returns the probability of an email being classified as spam (yes), which is then categorized based on a 0.5 threshold. Subsequently, we evaluate the classification performance using the Confusion Matrix, and compute the ROC Curve and AUC value to assess the overall classification capability of the model.

```
varImpPlot(rf_model)
```

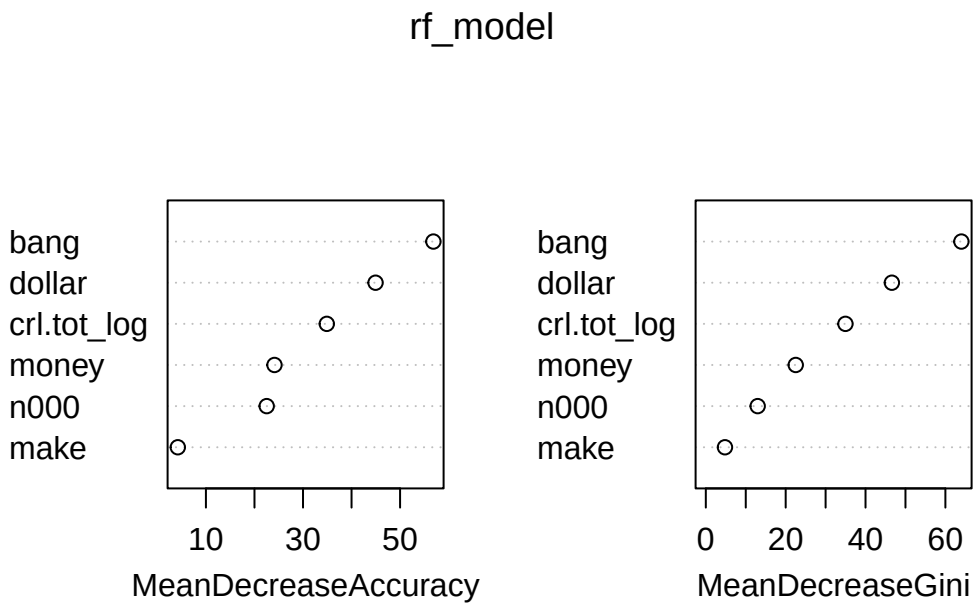

Figure 21: RF Variable Importance Ranking

Figure 21 represents the Variable Importance evaluation of the Random Forest model, illustrating the impact of different features on the model.Left plot (MeanDecreaseAccuracy) indicates that the bang variable is the most important, as its removal results in the greatest decrease in model accuracy. crl.tot_log and dollar are also significant, having a considerable impact. The make variable has a minor influence, contributing the least to the model. Right plot (MeanDecreaseGini) shows that bang remains the most important variable, making the largest contribution to Gini-based splits. dollar and crl.tot_log also have a strong influence, meaning they effectively distinguish between the yesno categories. The make variable has the lowest Gini importance, indicating it is used less frequently in the splitting process.

```
get_model_metrics <- function(model_name, confusion,k) {
  data.frame(
    Model = model_name,
    Accuracy = confusion$overall["Accuracy"],
    Sensitivity = confusion$byClass["Sensitivity"],
    Specificity = confusion$byClass["Specificity"],
    Precision = confusion$byClass["Precision"],
    AUC=as.numeric(k$auc),
```

```
    stringsAsFactors = FALSE
  )
}
results <- bind_rows(
  get_model_metrics("Random Forest",
                    confusion = rf_confusion,k=rf_roc),
  get_model_metrics("GLM",
                    confusion = glm_confusion,k=glm_roc)
) %>%
  mutate(across(-Model, ~ round(., 3)))

knitr::kable(results, align = "c")
```

	Model	Accuracy	Sensitivity	Specificity	Precision	AUC
Accuracy…1	Random Forest	0.879	0.925	0.782	0.898	0.934
Accuracy…2	GLM	0.870	0.938	0.731	0.878	0.916

In this step, we calculate and compare the classification performance of Random Forest (RF) and Logistic Regression (GLM). Using the Confusion Matrix and AUC (Area Under the ROC Curve), we extract Accuracy, Sensitivity, Specificity, and Precision, then format the results into a table for an intuitive comparison of different models' performance.

```
plot(glm_roc, col = "blue")
lines(rf_roc, col = "red")
legend("bottomright", legend = c("GLM", "randomForest"), col = c("blue", "red"), lwd = 2)
```

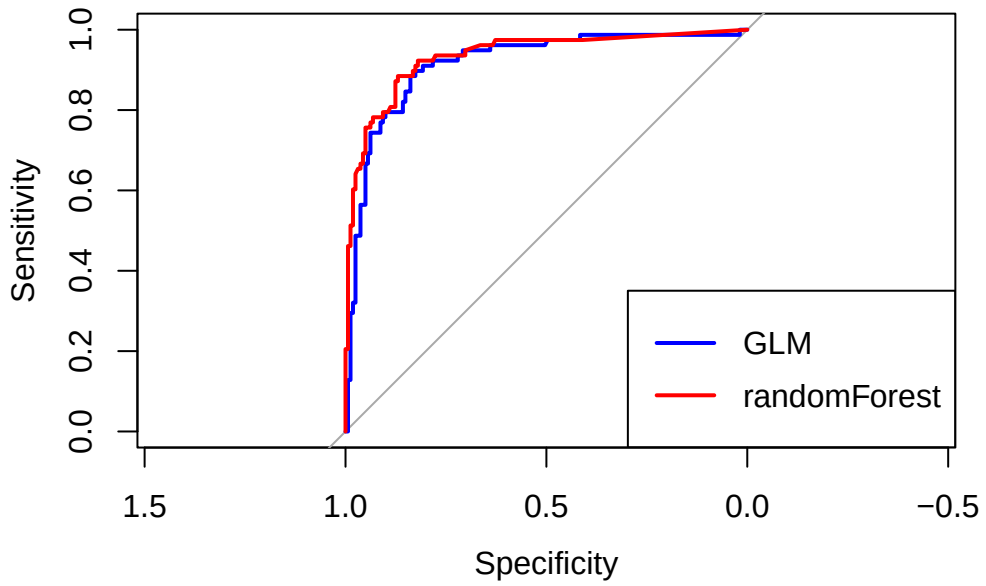

Figure 22: ROC Curve Comparison (GLM & RF)

Figure 22 displays the ROC curves for GLM (Logistic Regression, blue) and Random Forest (red) in a binary classification task. The two curves, GLM (blue) and Random Forest (red), are very close to each other, indicating that both models have similar classification performance. The Random Forest (red) slightly outperforms GLM in certain areas, but the overall difference is minimal.