How to classify an eamil as spam

Group 25: Hang Cheng, Haowei Yan, Weijan Li, Wenli Lyu, Zehao Wang

1 Introductions

Spam has caused some distress in people's daily life, so identifying spam correctly becomes more and more important nowadays. This study aims to finding Which text characteristics influence whether an email will be classified as spam or not by analyzing the data shared with the UCI Machine Learning Repository.

2 Data Reading

```
# Load the necessary package
library(tidyverse)
library(moderndive)
library(gapminder)
library(sjPlot)
library(stats)
library(jtools)
library(skimr)
library(pROC)
library(ResourceSelection)
```

```r
# Read CSV data
d25 <- read.csv("dataset25.csv")
```

2.1 Summary of the Data

```r
# Generate a summary of the dataset
d25 %>% skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	921
Number of columns	7
Column type frequency:	
character	1
numeric	6
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
yesno	0	1	1	1	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
crl.tot	0	1	275.76	491.47	1	41	102.00	267.00	3752.00	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
dollar	0	1	0.08	0.28	0	0	0.00	0.06	5.30	
bang	0	1	0.29	0.88	0	0	0.04	0.33	19.13	
money	0	1	0.08	0.32	0	0	0.00	0.00	6.66	
n000	0	1	0.11	0.40	0	0	0.00	0.00	5.45	
make	0	1	0.11	0.30	0	0	0.00	0.00	2.77	

Summary of the Data

```
# Select relevant variables and transform data
d25.spam <- d25 %>%
  select(yesno, crl.tot, dollar, bang, money, n000, make)

# Convert 'yesno' to a factor and scale 'crl.tot'
d25.spam$yesno <- as.factor(d25.spam$yesno)
d25.spam$crl.tot <- d25.spam$crl.tot / 100
```

According to the data, six main characteristics may exert an influence on classifying an email as spam. We divide “crl.tot” by 100 because the number is much larger than other data.

2.2 Data Visualization

```
# Create histograms for selected variables
d25.spam %>%
  pivot_longer(cols = c(crl.tot, dollar, bang, money, n000, make),
               names_to = "variable", values_to = "value") %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 30, fill = "blue", alpha = 0.5) +
  facet_wrap(~variable, scales = "free") +
  theme_minimal()
```

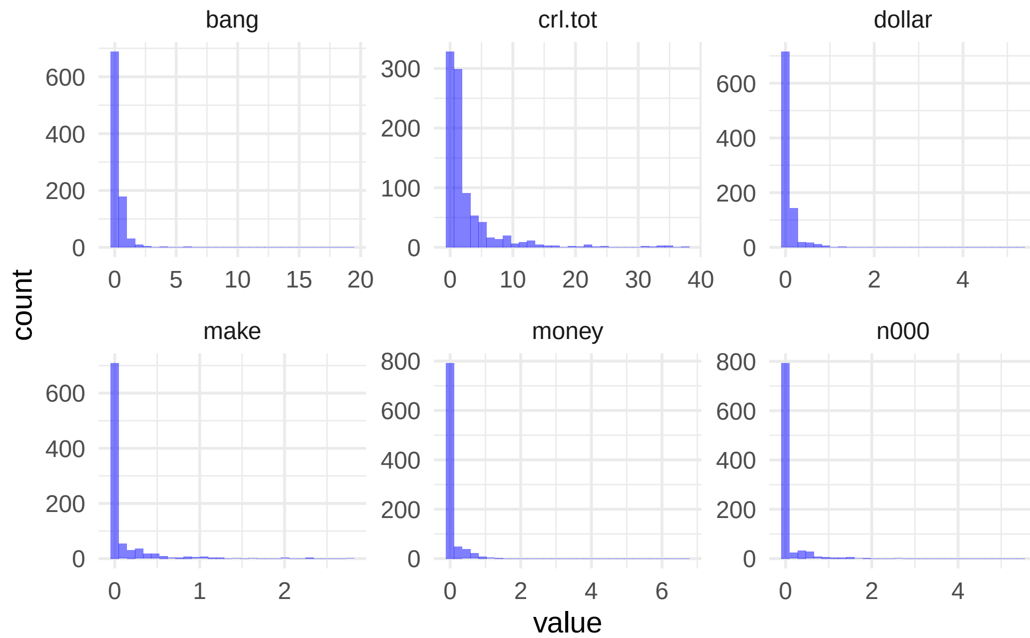


Figure 1: Histogram of Variables

```
# Create boxplots for selected variables grouped by spam label
d25.spam %>%
  pivot_longer(cols = c(crl.tot, dollar, bang, money, n000, make),
               names_to = "variable", values_to = "value") %>%
  ggplot(aes(x = yesno, y = value, fill = yesno)) +
  geom_boxplot() +
  facet_wrap(~variable, scales = "free") +
  theme_minimal() +
  theme(legend.position = "none")
```

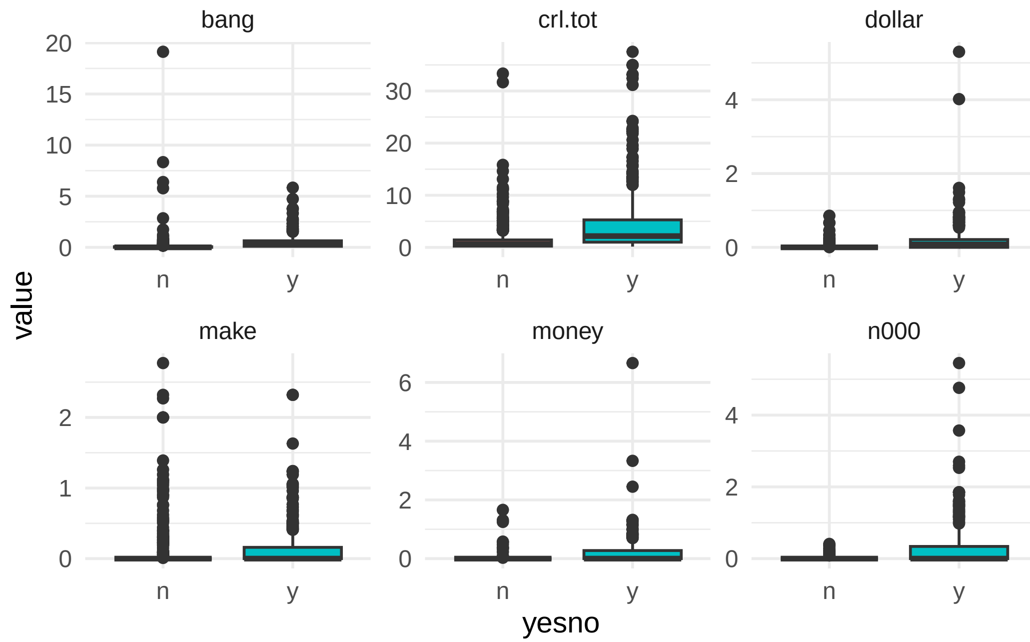


Figure 2: Boxplot of Variables by Spam Label

The distribution of the explanatory variables and their skewness can be seen in these plots, with a large number of discrete points that may need to be further analyzed and processed.

3 Analysis of Six Main Characteristics

We firstly analyze these characteristics separately.

3.1 Total length of uninterrupted sequences of capitals

```
# Boxplot of 'crl.tot' grouped by spam label
ggplot(data = d25.spam, aes(x = yesno, y = crl.tot, fill = yesno)) +
  geom_boxplot() +
  labs(x = "Is the email a spam?") +
  theme_minimal() +
  theme(legend.position = "none")
```

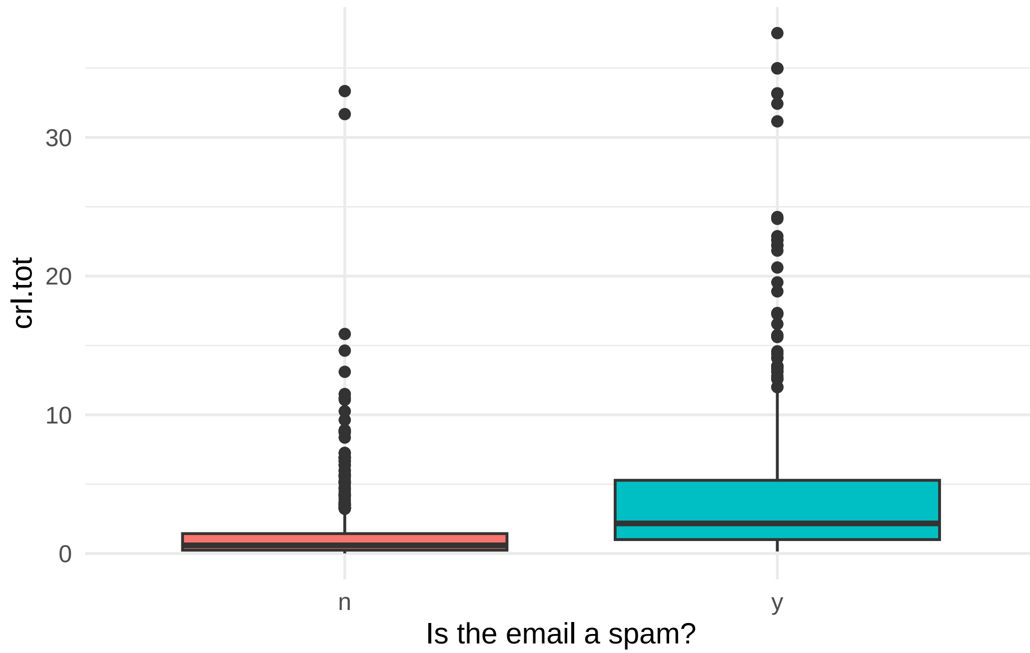


Figure 3: Total length of uninterrupted sequences of capitals in an email

The boxplot shows that, on average, there are more uninterrupted sequences of capitals in a spam than in a normal email.

3.2 Occurrences of the dollar sign

```
# Boxplot of 'dollar' grouped by spam label
ggplot(data = d25.spam, aes(x = yesno, y = dollar, fill = yesno)) +
  geom_boxplot() +
  labs(x = "Is the email a spam?") +
  theme_minimal() +
  theme(legend.position = "none")
```

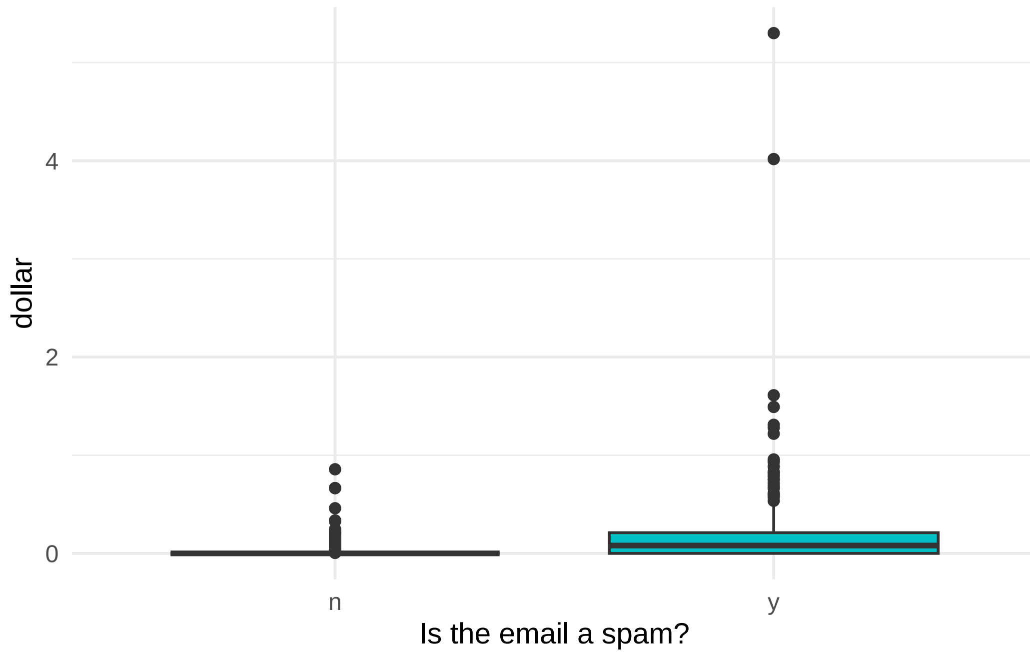


Figure 4: Occurrences of the dollar sign in an email

This graph shows that, on average, dollar sign appears more frequently in a spam.

3.3 Occurrences of '!'

```
# Boxplot of 'bang' grouped by spam label
ggplot(data = d25.spam, aes(x = yesno, y = bang, fill = yesno)) +
  geom_boxplot() +
  labs(x = "Is the email a spam?") +
  theme_minimal() +
  theme(legend.position = "none")
```

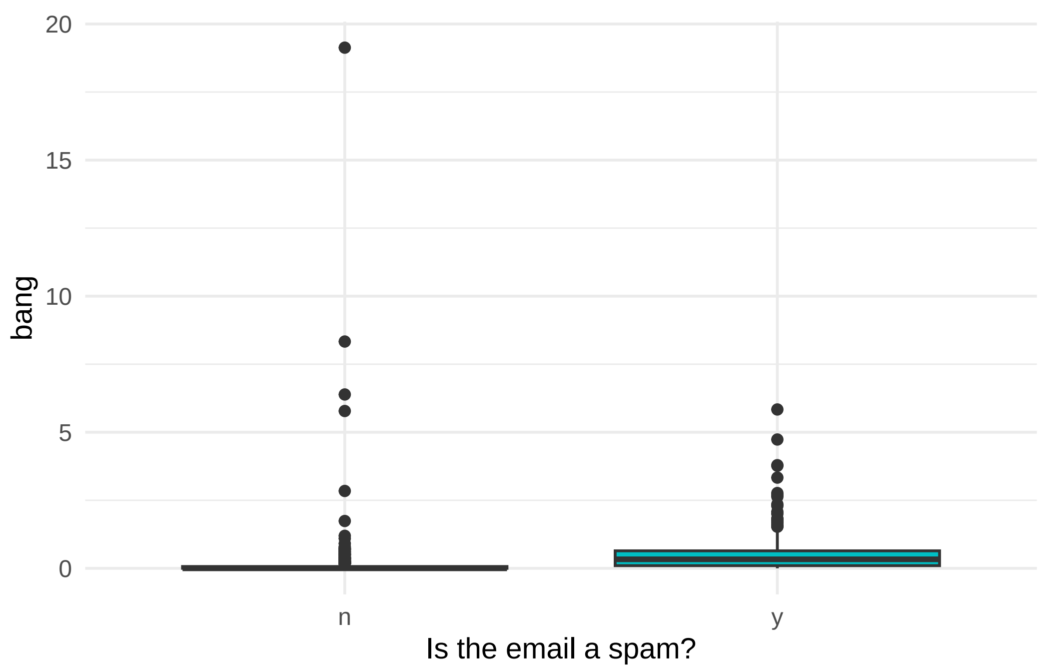


Figure 5: Occurrences of '!' in an email

This boxplot shows that, on average, exclamation mark tend to occur more in a spam.

3.4 Occurrences of 'money'

```
# Boxplot of 'money' grouped by spam label
ggplot(data = d25.spam, aes(x = yesno, y = money, fill = yesno)) +
  geom_boxplot() +
  labs(x = "Is the email a spam?") +
  theme_minimal() +
  theme(legend.position = "none")
```

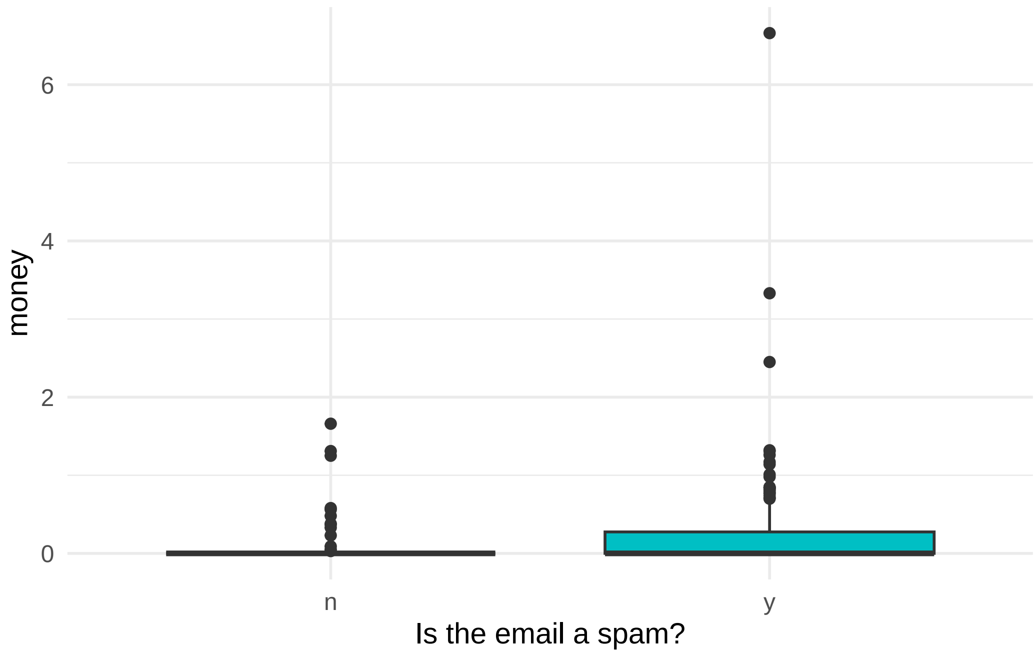


Figure 6: Occurrences of 'money' in an email

The graph shows that, on average, 'money' appears more frequently in a spam.

3.5 Occurrences of the string '000'

```
# Boxplot of 'n000' grouped by spam label
ggplot(data = d25.spam, aes(x = yesno, y = n000, fill = yesno)) +
  geom_boxplot() +
  labs(x = "Is the email a spam?") +
  theme_minimal() +
  theme(legend.position = "none")
```

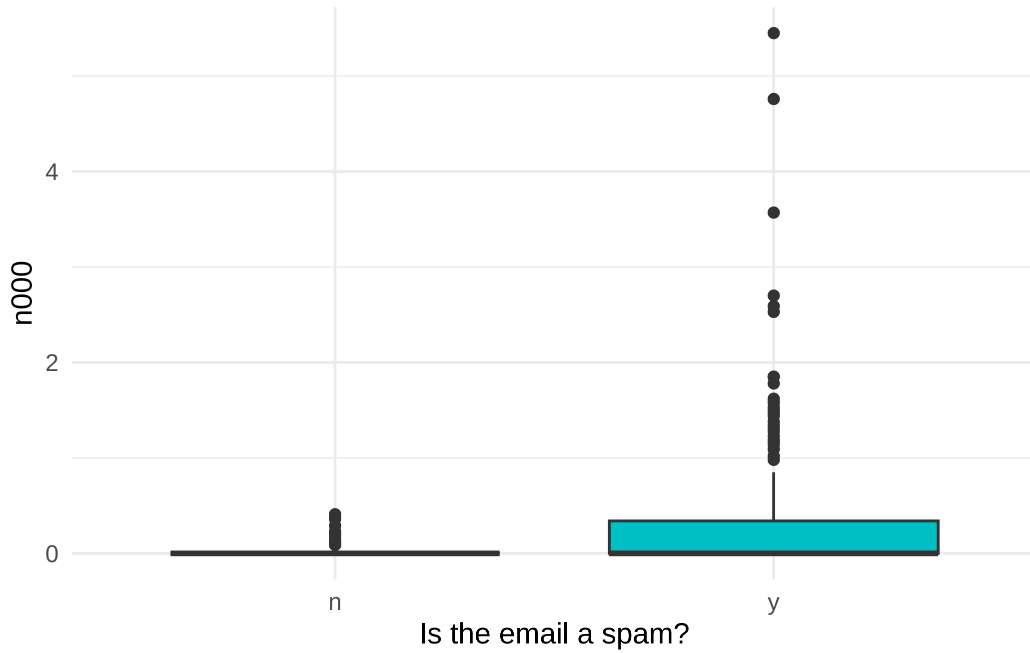


Figure 7: Occurrences of the string '000' in an email

The boxplot shows that, on average, the string '000' is more likely to occur in a spam.

3.6 Occurrences of 'make'

```
# Boxplot of 'make' grouped by spam label
ggplot(data = d25.spam, aes(x = yesno, y = make, fill = yesno)) +
  geom_boxplot() +
  labs(x = "Is the email a spam?") +
  theme_minimal() +
  theme(legend.position = "none")
```

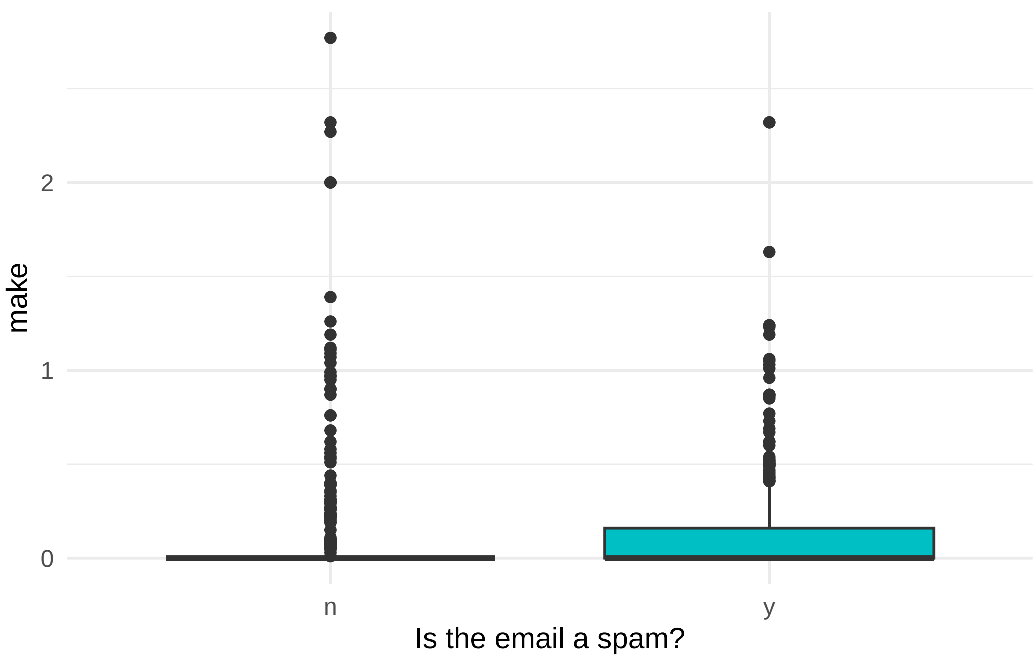


Figure 8: Occurrences of 'make' in an email

This graph shows that, on average, the occurences of 'make' in s spam is slightly more than in a noraml email.

4 Regression Results of the Data by using Generalized Linear Models

4.1 Fitting the full model

```r
# Fit a logistic regression model predicting spam emails
model.spam <- glm(yesno ~ crl.tot + dollar + bang + money + n000 + make,
                  data = d25.spam,
                  family = binomial(link = "logit"))

# Display model summary
model.spam %>%
  summary()
```

```
Call:
glm(formula = yesno ~ crl.tot + dollar + bang + money + n000 +
    make, family = binomial(link = "logit"), data = d25.spam)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.71120    0.12279 -13.936  < 2e-16 ***
crl.tot      0.13173    0.02912   4.523 6.09e-06 ***
dollar       6.96046    1.21199   5.743 9.30e-09 ***
bang         0.73018    0.18206   4.011 6.05e-05 ***
money        3.51853    0.69462   5.065 4.08e-07 ***
n000         5.50500    1.20129   4.583 4.59e-06 ***
make         0.01348    0.30804   0.044    0.965
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 1241.77  on 920  degrees of freedom
Residual deviance:  809.54  on 914  degrees of freedom
AIC: 823.54

Number of Fisher Scoring iterations: 7
```

The spam classification model is defined as:

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

Where:

- crl.tot: Total length of uninterrupted capital sequences
- dollar: Frequency of the dollar sign (`$`)
- bang: Frequency of exclamation marks (`!`)
- money: Frequency of the word "money"
- n000: Frequency of the string "000"
- make: Frequency of the word "make"

The coefficients of six characteristics are all positive, suggesting that spam tends to have more of these text characteristics. All the coefficients of the characteristics, except 'make', are significant because of the low p-values. But there is a warning message shows that glm.fit: fitted probabilities numerically 0 or 1 occurred. And the distributions of many of the explanatory variables were heavily skewed, so we decided to treat the data.

4.2 Transformation of data

Crl.tot shows a right-skewed distribution (mean 2.758, maximum 37.52), but there is a high proportion of non-zero values, which is suitable for mitigating the skewness by logarithmic transformation. Bang is heavily right skewed (mean 0.292, max 19.13), but has a certain percentage of non-zero values (median 0.044), which is suitable for logarithmic transformation.

```
# Apply log transformation to selected variables
d25.spam$log_crl.tot <- log(d25.spam$crl.tot + 1)
d25.spam$log_bang <- log(d25.spam$bang + 1)
```

Most of the values of dollar , money, n000 and make are 0, with more extreme values, and the model can be simplified by binning to reduce noise and nonlinear effects.

```
# Bin numeric variables into categorical groups
d25.spam$dollar_bin <- cut(d25.spam$dollar,
                           breaks = c(-1, 0, 0.1, Inf),
                           labels = c("0", "low", "high"))
d25.spam$money_bin <- cut(d25.spam$money,
                          breaks = c(-1, 0, 0.1, Inf),
                          labels = c("0", "low", "high"))
d25.spam$n000_bin <- cut(d25.spam$n000,
                         breaks = c(-1, 0, 0.1, Inf),
                         labels = c("0", "low", "high"))
d25.spam$make_bin <- cut(d25.spam$make,
                         breaks = c(-1, 0, 0.1, Inf),
                         labels = c("0", "low", "high"))
```

4.3 Visualization of processed data

```
# Visualizing log-transformed variables
p1 <- ggplot(d25.spam, aes(x = log_crl.tot, fill = yesno)) +
  geom_density(alpha = 0.6) +
  labs(title = "Distribution of log(crl.tot + 1)", x = "log(crl.tot + 1)", y = "Density") +
  theme_minimal()

p2 <- ggplot(d25.spam, aes(x = yesno, y = log_crl.tot, fill = yesno)) +
```

```r
  geom_boxplot() +
  labs(title = "log(crl.tot + 1) by Spam Class", x = "Spam Class", y = "log(crl.tot + 1)") +
  theme_minimal()

p3 <- ggplot(d25.spam, aes(x = log_bang, fill = yesno)) +
  geom_density(alpha = 0.6) +
  labs(title = "Distribution of log(bang + 1)", x = "log(bang + 1)", y = "Density") +
  theme_minimal()

p4 <- ggplot(d25.spam, aes(x = yesno, y = log_bang, fill = yesno)) +
  geom_boxplot() +
  labs(title = "log(bang + 1) by Spam Class", x = "Spam Class", y = "log(bang + 1)") +
  theme_minimal()

p1; p2; p3; p4;
```

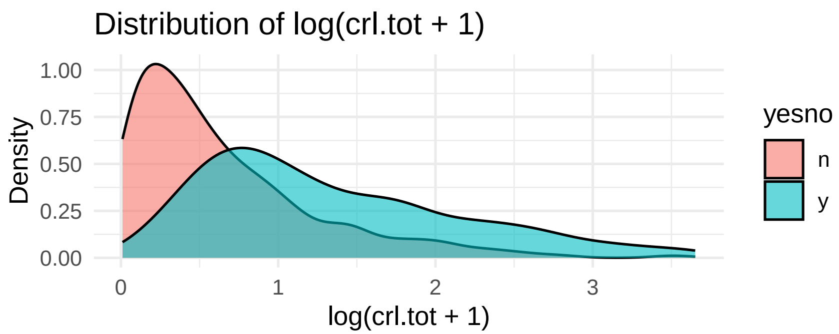


Figure 9: Group 1: Transformed Variables

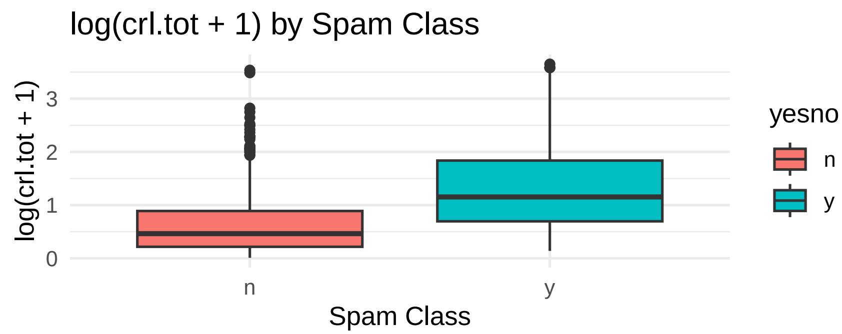


Figure 10: Group 1: Transformed Variables

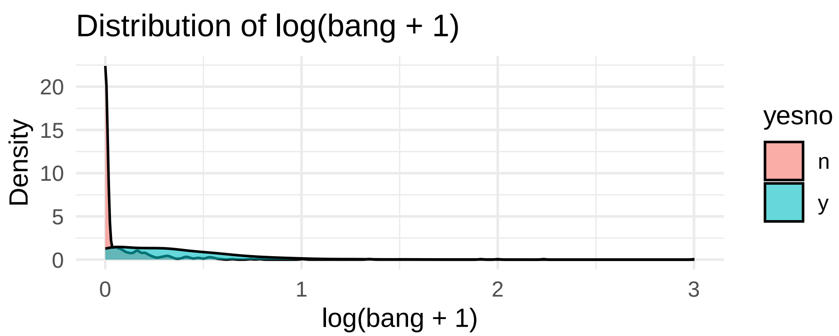


Figure 11: Group 1: Transformed Variables

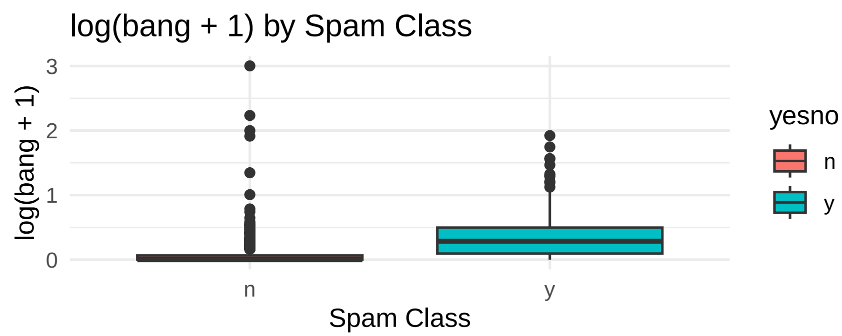


Figure 12: Group 1: Transformed Variables

The log-transformed variables (crl.tot and bang) show distinct differences between spam (y) and non-spam (n) emails. The density plots indicate that spam emails tend to have higher values for both log(crl.tot + 1) and log(bang + 1). The boxplots further confirm this trend, showing a higher median and broader distribution for spam emails, particularly for log(bang + 1), which has many extreme values.

```
# Visualizing binned variables
p5 <- ggplot(d25.spam, aes(x = dollar_bin, fill = yesno)) +
  geom_bar(position = "fill") +
  labs(title = "Dollar Frequency Bins vs Spam", x = "Dollar Bin", y = "Proportion") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

p6 <- ggplot(d25.spam, aes(x = money_bin, fill = yesno)) +
  geom_bar(position = "fill") +
  labs(title = "Money Frequency Bins vs Spam", x = "Money Bin", y = "Proportion") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

p7 <- ggplot(d25.spam, aes(x = n000_bin, fill = yesno)) +
  geom_bar(position = "fill") +
  labs(title = "n000 Frequency Bins vs Spam", x = "n000 Bin", y = "Proportion") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

p8 <- ggplot(d25.spam, aes(x = make_bin, fill = yesno)) +
  geom_bar(position = "fill") +
  labs(title = "Make Frequency Bins vs Spam", x = "Make Bin", y = "Proportion") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

p5; p6; p7; p8;
```

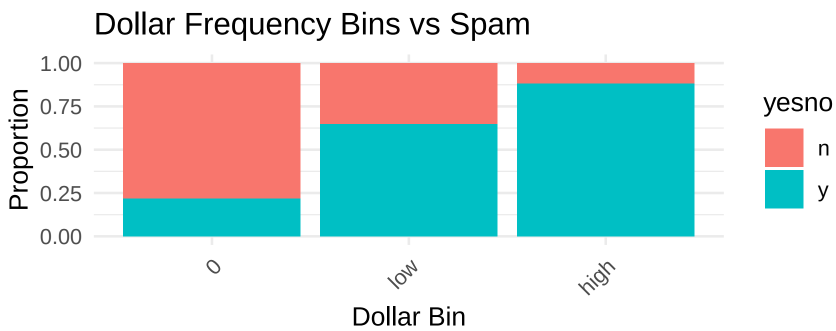


Figure 13: Group 2: Binned Variables

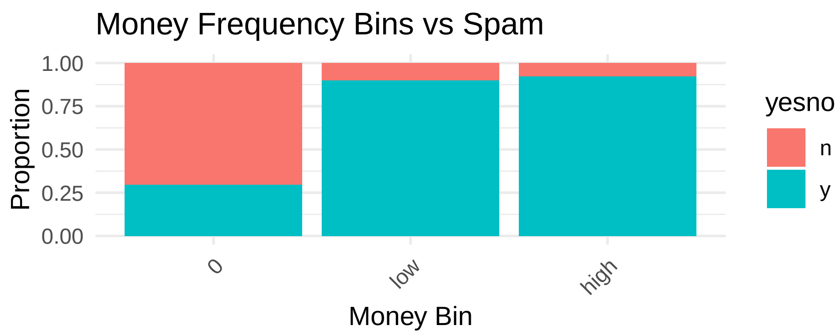


Figure 14: Group 2: Binned Variables

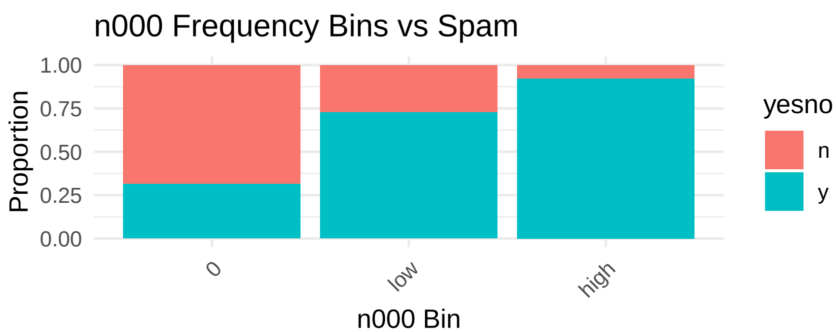


Figure 15: Group 2: Binned Variables

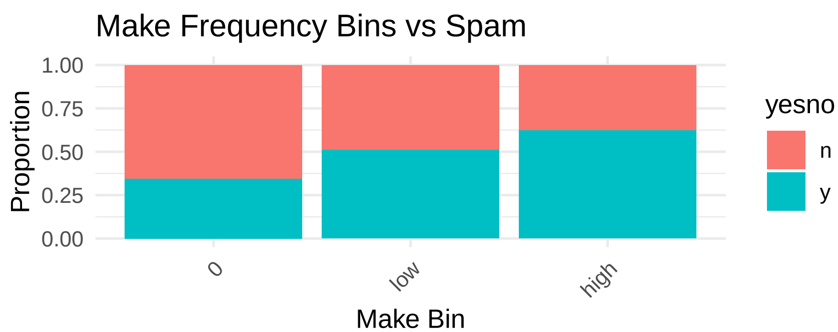


Figure 16: Group 2: Binned Variables

The proportion plots reveal strong associations between categorical frequency bins and spam classification. Emails with high occurrences of "$" (dollar), "money," and "000" have a much greater proportion of spam, suggesting that these words are strong spam indicators. Conversely, the presence of the word "make" does not show a clear spam association, as its proportions remain more balanced across spam and non-spam emails.

4.4 Fitting a model with processed data

```r
# Fit a second logistic regression model with transformed variables
model.spam2 <- glm(yesno ~ log_crl.tot + dollar_bin + log_bang + money_bin + n000_bin + make_bin,
                   data = d25.spam,
                   family = binomial(link = "logit"))

# Display model summary
model.spam2 %>%
  summary()
```

```
Call:
glm(formula = yesno ~ log_crl.tot + dollar_bin + log_bang + money_bin +
    n000_bin + make_bin, family = binomial(link = "logit"), data = d25.spam)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     -2.9055     0.2069 -14.041  < 2e-16 ***
log_crl.tot      1.1687     0.1855   6.299 2.99e-10 ***
dollar_binlow    1.0821     0.3575   3.027 0.002471 **
dollar_binhigh   2.0136     0.2933   6.866 6.60e-12 ***
log_bang         3.6799     0.4226   8.709  < 2e-16 ***
money_binlow     0.9934     0.8198   1.212 0.225599
money_binhigh    2.1536     0.3877   5.555 2.78e-08 ***
n000_binlow     -0.7254     1.0385  -0.699 0.484855
n000_binhigh     1.5668     0.4373   3.583 0.000340 ***
make_binlow     -2.2436     0.6048  -3.710 0.000208 ***
make_binhigh    -0.3932     0.3174  -1.239 0.215468
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 1241.77  on 920  degrees of freedom
Residual deviance:  660.07  on 910  degrees of freedom
AIC: 682.07

Number of Fisher Scoring iterations: 6
```

The refined logistic regression model is defined as:

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

Where:

- $\log(\text{crl.tot}+1)$: Log-transformed total length of capital sequences.
- dollar_bin: Binned frequency of `$` (categories: `0`, `low`, `high`).
- $\log(\text{bang}+1)$: Log-transformed frequency of `!`.
- money_bin, n000_bin, make_bin: Binned frequencies of "money", "000", and "make" (categories: `0`, `low`, `high`).

This model does not have the warning messages that appear in the full model. The Longer sequences of capital letters (log_crl.tot) and frequent exclamation marks (log_bang) strongly increase spam likelihood, with highly significant coefficients ($p < 0.001$). High-frequency dollar signs (dollar_binhigh) and mentions of "money" (money_binhigh) are also significant spam indicators. Notably, even low-frequency dollar signs (dollar_binlow) show a moderate positive effect. The presence of "000" strings (n000_binhigh) further raises spam risk. Conversely, low-frequency use of "make" (make_binlow) significantly reduces spam probability. Variables like money_binlow, n000_binlow, and make_binhigh are statistically insignificant ($p > 0.05$), suggesting limited impact.

We chose to merge certain variable categories (e.g., combining "low" and "0" frequency bins) to address statistical insignificance while preserving meaningful information.

4.5 Combining insignificant variables and fitting a new model

```
# Merge categories for selected binned variables
d25.spam <- d25.spam %>%
  mutate(
    money_bin_merged = case_when(
      money_bin %in% c("0", "low") ~ "0_low",
      money_bin == "high" ~ "high"
    ),
    n000_bin_merged = case_when(
      n000_bin %in% c("0", "low") ~ "0_low",
      n000_bin == "high" ~ "high"
    ),
    make_bin_merged = case_when(
      make_bin == "0" ~ "0",
      make_bin %in% c("low", "high") ~ "present"
    )
  )

# Fit a third logistic regression model with merged categories
model.spam3 <- glm(yesno ~ log_crl.tot + dollar_bin + log_bang + money_bin_merged + n000_bin_merged + make_bin_merged,
                   data = d25.spam,
                   family = binomial(link = "logit"))

# Display model summary
model.spam3 %>%
  summary()
```

```
Call:
glm(formula = yesno ~ log_crl.tot + dollar_bin + log_bang + money_bin_merged +
    n000_bin_merged + make_bin_merged, family = binomial(link = "logit"),
    data = d25.spam)
```

```
Coefficients:
                       Estimate Std. Error z value Pr(>|z|)    
(Intercept)             -2.7898     0.1952 -14.294  < 2e-16 ***
log_crl.tot              1.0157     0.1700   5.976 2.29e-09 ***
dollar_binlow            0.8267     0.3364   2.458    0.014 *  
dollar_binhigh           2.0464     0.2896   7.067 1.59e-12 ***
log_bang                 3.6947     0.4186   8.825  < 2e-16 ***
money_bin_mergedhigh     2.2422     0.3835   5.847 5.02e-09 ***
n000_bin_mergedhigh      1.7653     0.4094   4.312 1.62e-05 ***
make_bin_mergedpresent  -0.7339     0.2987  -2.457    0.014 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1241.77  on 920  degrees of freedom
Residual deviance:  671.07  on 913  degrees of freedom
AIC: 687.07

Number of Fisher Scoring iterations: 6
```

The final logistic regression model with merged bins is defined as:

$$\begin{aligned}
\log\left(\frac{P(\text{yesno} = \text{spam})}{1 - P(\text{yesno} = \text{spam})}\right) = \beta_0 &+ \beta_1 \cdot \log(\text{crl.tot} + 1) + \beta_2 \cdot \text{dollar_bin} \\
&+ \beta_3 \cdot \log(\text{bang} + 1) + \beta_4 \cdot \text{money_bin_merged} \\
&+ \beta_5 \cdot \text{n000_bin_merged} + \beta_6 \cdot \text{make_bin_merged}
\end{aligned}$$

Where:

- $\log(\text{crl.tot} + 1)$: Log-transformed total capital sequence length.

- dollar_bin: Binned `$` frequency (`0`, `low`, `high`).
- log(bang + 1): Log-transformed `!` frequency.
- money_bin_merged: Merged bins for "money" (`0_low`, `high`).
- n000_bin_merged: Merged bins for "000" (`0_low`, `high`).
- make_bin_merged: Merged bins for "make" (`0`, `present`).

The refined model demonstrates strong statistical performance with all retained variables achieving significance at $\alpha = 0.05$ or stricter thresholds, indicating strong predictors of spam classification. The AIC (687.07) remains nearly unchanged compared to the previous model (AIC: 682.07), suggesting minimal information loss despite reduced complexity.

5 Assess the Model

5.1 Assess the predictive power

```r
# Compute predicted probabilities for each model
predicted_prob <- predict(model.spam, type = "response")
predicted_prob2 <- predict(model.spam2, type = "response")
predicted_prob3 <- predict(model.spam3, type = "response")

# Compute ROC curves
roc_obj <- roc(response = d25.spam$yesno, predictor = predicted_prob)
roc_obj2 <- roc(response = d25.spam$yesno, predictor = predicted_prob2)
roc_obj3 <- roc(response = d25.spam$yesno, predictor = predicted_prob3)

# Plot ROC curves for each model
plot(roc_obj, main = "ROC Curve for Spam Detection Model 1", print.auc = TRUE, auc.polygon = TRUE, legacy.axes = TRUE)
```

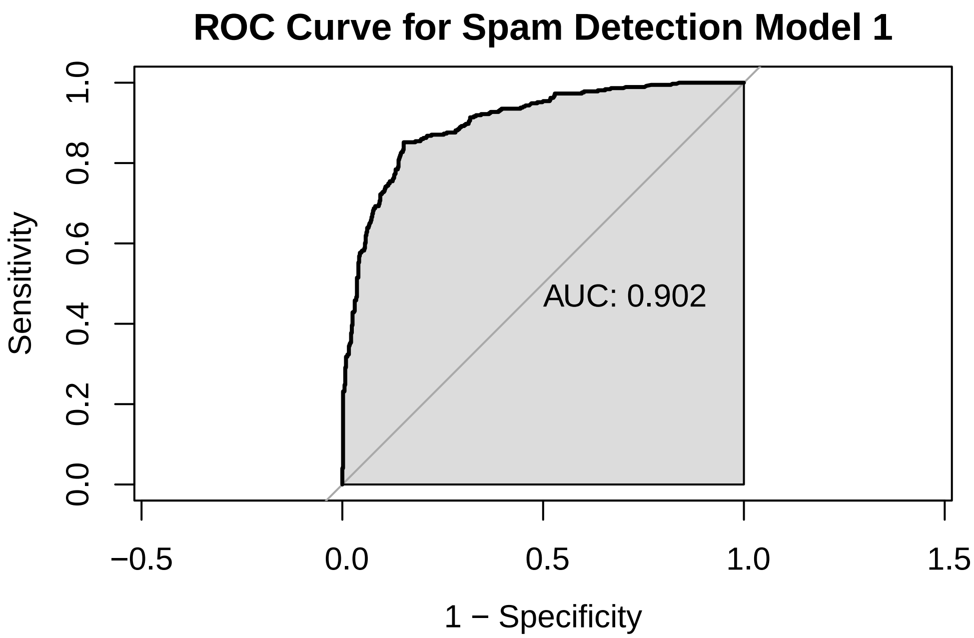


Figure 17: ROC Curves for Spam Detection Models

```
plot(roc_obj2, main = "ROC Curve for Spam Detection Model 2", print.auc = TRUE, auc.polygon = TRUE, legacy.axes = TRUE)
```

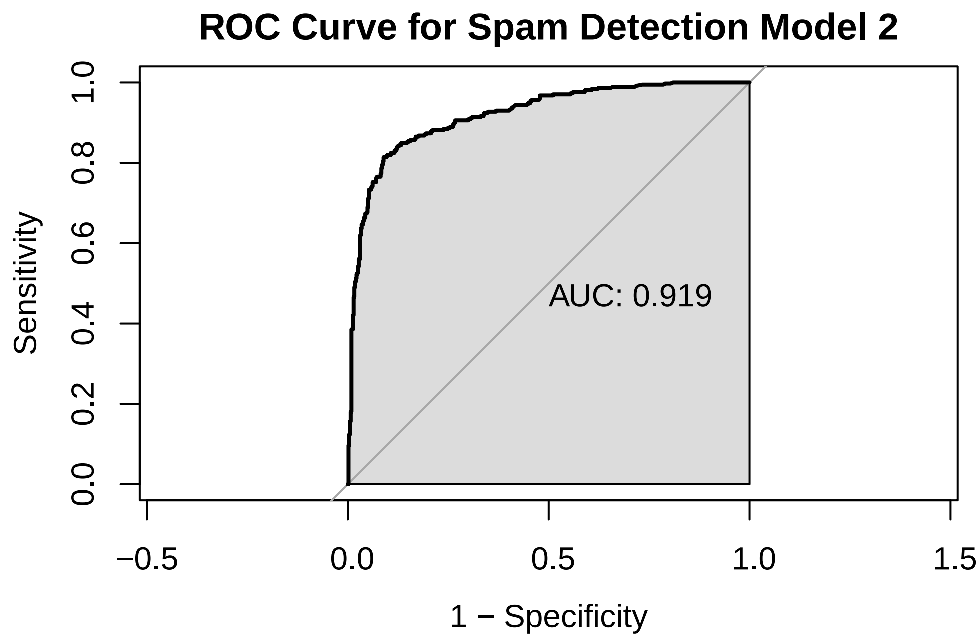


Figure 18: ROC Curves for Spam Detection Models

```
plot(roc_obj3, main = "ROC Curve for Spam Detection Model 3", print.auc = TRUE, auc.polygon = TRUE, legacy.axes = TRUE)
```

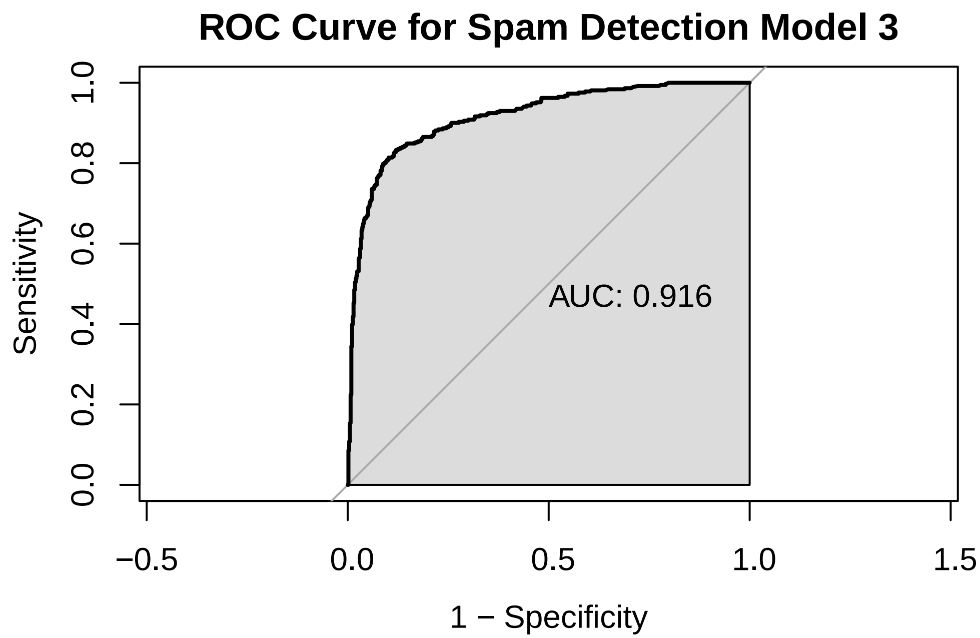


Figure 19: ROC Curves for Spam Detection Models

```
# Compute and display AUC values
auc_value <- auc(roc_obj)
auc_value2 <- auc(roc_obj2)
auc_value3 <- auc(roc_obj3)
cat("AUC1:", auc_value, "\n", "AUC2:", auc_value2, "\n", "AUC3:", auc_value3, "\n")
```

```
AUC1: 0.9015437
 AUC2: 0.9194609
 AUC3: 0.916197
```

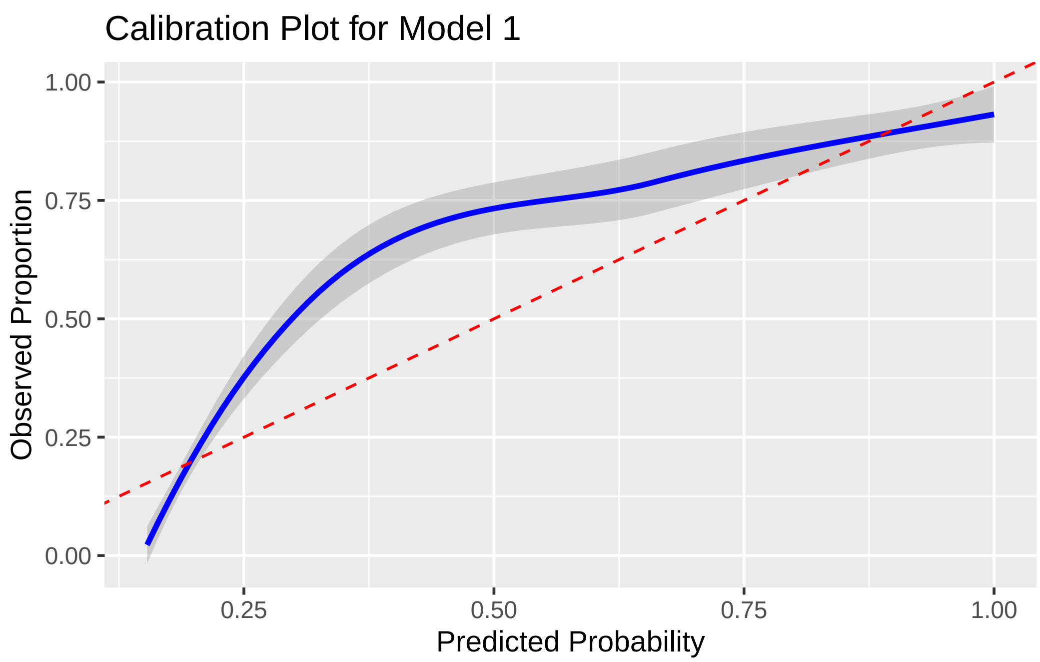
The three models all achieve an excellent AUC, indicating strong discriminatory power to distinguish spam from non-spam emails. The AUC of the second model is a little bit better than the AUC of the third model, but the difference is very small.

5.2 Hosmer-Lemeshow goodness of fit test

```r
# Convert spam labels to numeric for the test
d25.spam$yesno_numeric <- ifelse(d25.spam$yesno == "y", 1, 0)

# Hosmer-Lemeshow test for Model 1
hoslem.test(d25.spam$yesno_numeric, fitted(model.spam), g = 7)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  d25.spam$yesno_numeric, fitted(model.spam)
X-squared = 108.24, df = 5, p-value < 2.2e-16
```

```r
calibration_data <- data.frame(Predicted = predict(model.spam, type = "response"), Actual = d25.spam$yesno_numeric)

# Calibration plot for Model 1
ggplot(calibration_data, aes(x = Predicted, y = Actual)) +
  geom_smooth(color = "blue") +
  geom_abline(linetype = "dashed", color = "red") +
  labs(title = "Calibration Plot for Model 1", x = "Predicted Probability", y = "Observed Proportion")
```

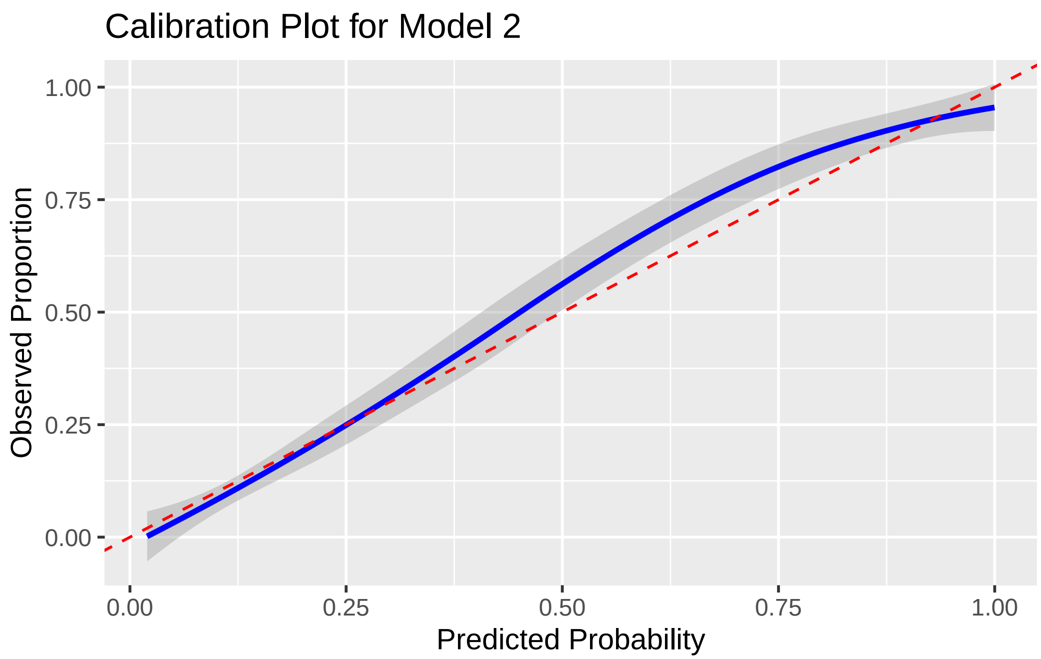



```
# Hosmer-Lemeshow test for Model 2
hoslem.test(d25.spam$yesno_numeric, fitted(model.spam2), g = 7)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  d25.spam$yesno_numeric, fitted(model.spam2)
X-squared = 13.75, df = 5, p-value = 0.01728
```

```
calibration_data2 <- data.frame(Predicted = predict(model.spam2, type = "response"), Actual = d25.spam$yesno_numeric)

# Calibration plot for Model 2
ggplot(calibration_data2, aes(x = Predicted, y = Actual)) +
  geom_smooth(color = "blue") +
```

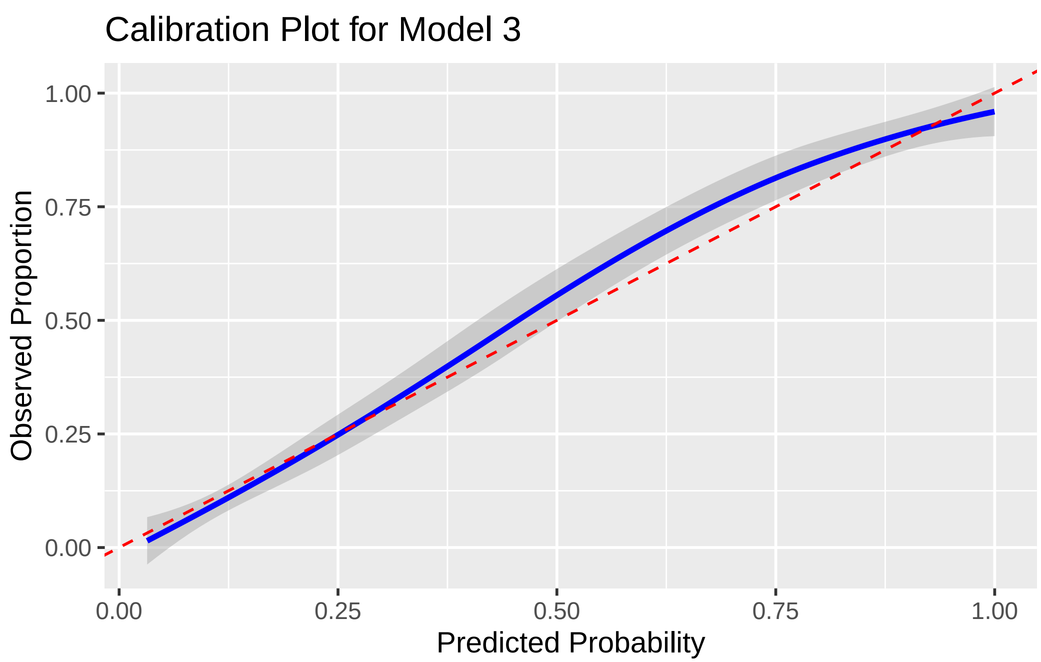
```r
  geom_abline(linetype = "dashed", color = "red") +
  labs(title = "Calibration Plot for Model 2", x = "Predicted Probability", y = "Observed Proportion")
```




```r
# Hosmer-Lemeshow test for Model 3
hoslem.test(d25.spam$yesno_numeric, fitted(model.spam3), g = 7)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  d25.spam$yesno_numeric, fitted(model.spam3)
X-squared = 10.621, df = 5, p-value = 0.05944
```

```r
calibration_data3 <- data.frame(Predicted = predict(model.spam3, type = "response"), Actual = d25.spam$yesno_numeric)

# Calibration plot for Model 3
ggplot(calibration_data3, aes(x = Predicted, y = Actual)) +
  geom_smooth(color = "blue") +
  geom_abline(linetype = "dashed", color = "red") +
  labs(title = "Calibration Plot for Model 3", x = "Predicted Probability", y = "Observed Proportion")
```



The Hosmer-Lemeshow test of the third model (p = 0.059) indicates borderline non-significant evidence of miscalibration, suggesting the model's predicted probabilities may slightly deviate from observed outcomes.

The calibration plot shows strong agreement between predicted and observed probabilities in low-to-mid ranges but reveals minor overestimation in high-risk predictions and slight underestimation at extreme probabilities, suggesting localized calibration biases.

The p-value for the first model is much less than $\alpha = 0.05$, and the calibration plot also shows very large calibration biases.

The second model also has a p-value of less than $\alpha = 0.05$, and the calibration plot also has more segments than the third model calibration biases.

6 Data Summary

```
# Visualizing the odds ratios from the final logistic regression model
plot_model(model.spam3, show.values = TRUE, title = "Odds Ratios for Model 3", show.p = FALSE, value.offset = 0.25) +
  theme_minimal()
```

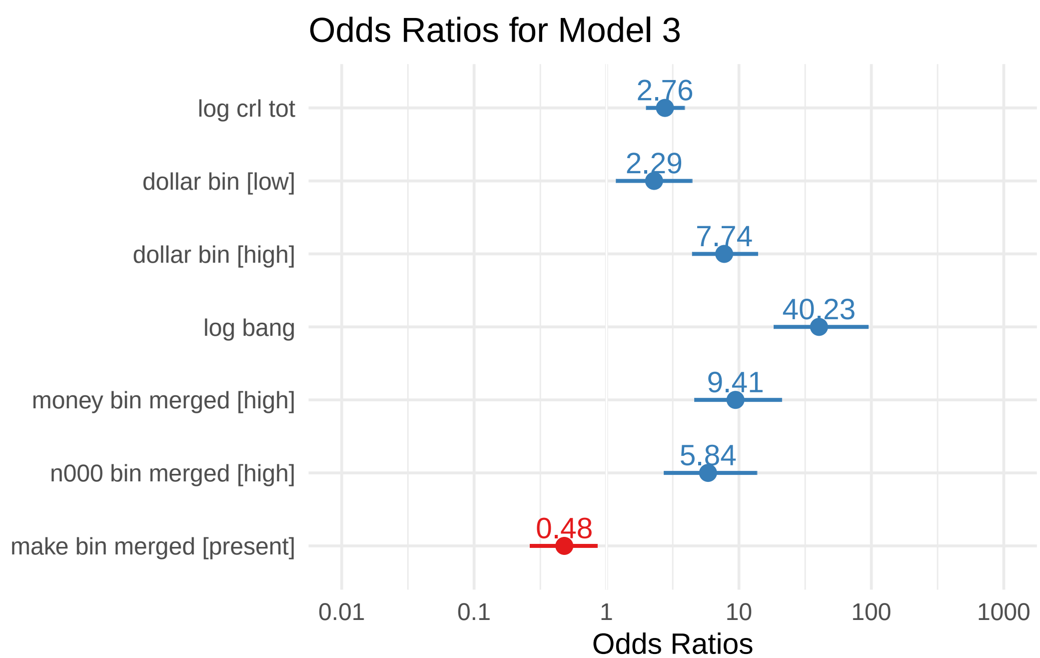


Figure 20: Odds Ratios for Model 3

According to the graph, a 1-unit increase in the log-transformed total length of capital letter sequences (log_crl.tot) increases spam odds by 176% (OR = 2.76). Emails with high-frequency dollar signs (dollar_binhigh) are 674% more likely to be spam (OR = 7.74), while low-frequency dollar signs (dollar_binlow) still elevate odds by 129% (OR = 2.29). Exclamation marks (log_bang) also exhibit the positive effect, with a 1-unit log increase raising spam likelihood by 3,923% (OR = 40.23). Mentions of "money" (OR = 9.41) and "000" (OR = 5.84) further amplify spam risk by 841% and 484%, respectively. Conversely, frequent use of "make" reduces spam odds by 52% (OR = 0.48), suggesting its association with legitimate content.

7 Conclusions

These findings highlight the importance of financial symbols ($, "money"), exaggerated punctuation (!), and anomalous patterns (capital bursts, "000") as spam indicators, while terms like "make" may signal non-spam context. This evidence directly informs targeted improvements for spam filtering systems.