Spam Filters

Omkar G. Kharkar

November 6th, 2011

Honor Pledge: On my honor, I pledge that I am the sole author of this paper and I have accurately cited all help and references used in its completion.

Summary:

In this report, I have included the analysis for optimal designs for spam filters, based on a Generalized Linear Model (GLM) filter and a time series filter for ham and spam data. The results, based on the graphs in Appendix A, show that two ARIMA functions, obtained by the use of the auto-ARIMA function, and a Stepwise Log-Transformed GLM are the best models when designing a spam filter.

1 Problem Description

1.1 Background

Spam is a colloquial expression that is used to describe junk e-mail. Schwartz & Garfinkel define it as "an unsolicited, unwanted message sent to you without permission. Spam is the internet's version of junk mail, telemarketing calls during dinner, crank phone calls, and leaflets posted around town, all rolled up into a single annoying electronic bundle" [1]. While spam is easily deleted by the receiver, it continues to pose two major problems for society: time and cost.

The time problem is again best summarized by Schwartz and Garfinkel:

> "Now it's true that each of these messages can be deleted with just a click of the mouse, which takes only three or four seconds: a few seconds to determine that the message is in fact spam plus a second to click Delete. But those seconds add up quickly: one million people clicking Delete corresponds to roughly a month of wasted human activity. Or put another way, if you get six spam messages a day, you're wasting two hours each year deleting spam" [1].

Cost is another problem facing large enterprises that operate hundreds of computers. Companies are becoming increasingly dependent on anti-spam technology, and must continually expand their employees' mailbox storage capacity to account for high volumes of suspicious e-mail. In a 2007 study conducted by Nucleus Research Inc., spam cost American companies $71 billion annually or $712/employee [2].

SoftScan, an e-mail security company, announced that "less than 3 percent of all the e-mail it scanned in December was legitimate" [3]. The increasing need for an effective spam filter along with the lab assignment [4] were both reasons for designing spam filters and this report.

1.2 Goal

The goal for designing spam filters in this report was to minimize the amount of spam messages received by a user. In developing the filters, the metric that was used to assess how effective the filters were was the probability of a message being marked as 'spam' based on specific criteria.

1.3 Data Source

The data for this report was obtained from a database by Reeber, Forman, and Suermondt [5], and contained information for 57 different categories. The 58^{th} variable was the response variable that indicated whether the message was classified as ham or spam. One problem that was identified with the data was the difference in years and quantity of the spam and ham. Spam was approx. 39% of the data, and was during the 1999 time period, different from ham, which was collected during the 2001 time period. Further studies should confirm the results obtained here by using several years of data to adjust for anomalies in the data.

2 EISE Approach to Spam Filter Design

My approach to the spam filter design has three components: a Static Filter Design, a Time Series Filter Design and an Integrated Filter Design. The subsections that follow describe each of these components.

2.1 Static Filter Design

The static filter design uses data in each email message to classify that message as spam or ham.

2.1.1 Hypothesis

For the static filter design, the hypotheses that I used were:

Null Hypothesis: Messages that are classified as spam do not have any significant correlation to any of the variables.

Alt. Hypothesis: Messages that are classified as spam do have significant correlation to at least 1 or more variable (s).

2.1.2 Visualization and Graphical Analysis

For this study, I used univariate and multivariate graphical methods for analysis. In order to gain a preliminary understanding of the data, a bar graph was used to identify the number of ham and spam messages. There were also several multivariate displays that were used for this report. First, the most important display was the Scatter Plot Matrix. These matrices display the correlations between variables and to the response variable. By observing the correlations between variables, this can indicate what overlap they have, and accordingly, I could choose which variables should be included or omitted in the generalized linear model. Principal components analysis was also used to understand the variance between the variables and to develop a principal components regression model as a potential spam filter design.

2.1.3 Generalized Linear Models (GLM)

There were several generalized linear models that were designed, and evaluated in this study. The models that were tested here were: Principal Components Regression, GLM with the 57 variables (GLM-57), Stepwise GLM, GLM-57 trained and tested, Stepwise GLM Trained & Tested (using the method of test sets), Log-Transformed GLM Trained & Tested, Stepwise Log-Transformed GLM. In order to compare the models, the Chi-Square test was used. To assess the model's predicted performance, the ROC (relative operating characteristic) curve was used to measure the probability of false positives vs. the probability of true positives (i.e. the informativeness of the forecast).

2.2 Time Series Filter Design

The time series filter uses counts of spam and ham at different time points to forecast the probability a message is spam at future times.

2.2.1 Hypothesis

For the Time Series Filter, the hypotheses I used were:

Hypothesis 1

Null Hypothesis – There is no serial correlation between ham messages.
Alt. Hypothesis - There is a serial correlation between the ham and spam messages.

Hypothesis 2

Null Hypothesis – There is no serial correlation between spam messages.
Alt. Hypothesis – There is a serial correlation between the ham and spam messages.

2.2.2 Visualization and Graphical Analysis

In order to develop the models here, the data was first divided into two distinct sets: ham and spam, each of which was analyzed separately to develop appropriate time series models. To visualize the data, ACF (Autocorrelation function) plots and PACF (Partial Autocorrelation function) plots were developed to understand the correlations and determine whether the time series was stationary or not. Then trend graphs and periodograms were used to understand if there was a distinct trend in the data, and whether seasonality was present. Finally, a forecast time-series was used to project what the expected value of spam would be 100 time units in the future.

2.2.3 Time Series Models

There were two major types of time series models that were used to obtain a time-series filter for the ham and spam messages. First, an ARIMA model was developed for the ham data using the first-order difference on the residuals to correct for stationarity and seasonality. The first-order differences corrected this problem, and an ARIMA(1,0,0) model was developed. Second, the Auto-ARIMA function was used to develop a computer-generated model of ARIMA(1,0,2) and ARIMA(3,0,2). The Box-Ljung statistic was used to verify whether the time series models developed were significant or not. The same procedures were applied to the spam time series data, but spam did not have a seasonality component, and as such, the residuals were not used. The models that were tested for spam were ARIMA(1,0,0), ARIMA(1,0,1) and ARIMA(3,0,2) here.

2.3 Integrated Filter Design

The integrated spam filter uses Bayes rule to combine the results from the static filter with the time series filter to produce an overall filter design.

$$P(Spam \mid Static\ Filter \bigcap Time\ Series\ Filter)$$
$$= \frac{P(Static\ Filter \bigcap Time\ Series\ Filter \mid Spam) * P(Spam)}{P(Static\ Filter \bigcap Time\ Series\ Filter)}$$
$$= \frac{P(Static\ Filter \mid Spam) * P(Time\ Series \mid Spam) * P(Spam)}{P(Static\ Filter) * P(Time\ Series\ Filter)}$$

$$P(Static\ Filter) * P(Time\ Series\ Filter)$$
$$= P(Static\ Filter \mid Ham) * P(Time\ Series\ Filter \mid Ham) * P(Ham)$$
$$+ P(Static\ Filter \mid Spam) * P(Time\ Series\ Filter \mid Spam) * P(Spam)$$

The mathematical model is shown above, for the Integrated filter design. This model helps to evaluate the question of what is the probability of a spam message given that the static filter and the time series filter are both being used? In the model above, the expressions "static filter" and "time series filter" refer to the probability of spam (or ham) determined by the static filter or the time series filter respectively. The formula can be solved by using the fact that the P(Static | Spam) can be calculated using the ROC graph, and the score table. Also, the P(Time Series | Spam) can be determined by calculating the proportion of spam / total mail and multiplying by all spam/total mail. (ham + spam).

3 Evidence

3.1 Static Filter Performance Results

This section gives the performance results for the components of the spam filter design.

3.1.1 Visualization and Graphical Analysis Results

A bar graph was obtained (Refer to Appendix, Figure 1) that showed the distribution of ham and spam. There was approximately twice as much ham in the dataset than spam. Second, the Scatter Plot matrices are shown (Figures 2 – 6) which show the correlations between the data set. Many of strong correlations between variables were noted for further analysis in the GLM models. Principal components analysis (Figures 7 - 8) was also used to determine which variables had corresponded most the Ham & Spam data set. Based on the principal components analysis, the variables that affect ham are very different from those that affect spam (Figure 7). As a result of this analysis, I was able to determine which variables are strongly correlated with one another, and this led to the idea that there distinct sets of principal components that represented spam and ham respectively. This was verified using PCA regression.

3.1.2 Generalized Linear Modeling Results

Several different types of models were designed in order to find the most effective GLM out of them. First, principal components regression was used (Figure 8) to determine whether there were distinct variables that would allow categorization of ham or spam respectively. This was shown be inferior to the GLM-57, based on the AIC score that was obtained. The GLM-57 and the stepwise models (Figure 9) were then developed, and the results compared using the Chi-square statistic (Figure 10). The Test-set models were then developed, and the above procedures were repeated to find the most accurate model. The stepwise trained & tested model (Figure 11) once again outperformed the trained & tested GLM and also had a more informative ROC curve (Figure 12). The Log-Transformed models were then develop and compared using the Chi-Square test statistic (Figure 13); the Stepwise Log-Transformed was a superior model. Then, the Stepwise GLM Trained & Tested and the Stepwise Log-Transformed GLM Trained & Tested were compared (Figure 14); the latter was shown to be the optimal GLM model out of all the models tested.

3.2 Time Series Filter Performance Results

3.2.1 Visualization and Graphical Analysis Results

The graphs that were created for the ham and spam time series respectively, allowed me to analyze the data for important characteristics, such as trend, seasonality and stationarity, and what further implications this would have. From the data, ham is shown to have a seasonality component, and a trend (Figures 16 -18). It is unclear whether the ham data is stationary (Figure 17), even though the ACF graph (Figure 18) decays exponentially and then varies sinusoidally. Therefore, I calculated the first-order difference, and also used the ACF, and PACF for the residuals to correct for the seasonality component (Figures 19 – 21). After using this transformation, the time series was then stationary.

The spam data did not have a periodic component, but did have a trend (Figure 29). Therefore the seasonality did not have to be removed here, and did not require the use of residuals. To correct for stationarity, the first-order difference was once again calculated, along with the ACF (Figures 31 - 32). The resulting data (Figure 32) was shown to be stationary.

3.2.2 Time Series Modeling Results

There were several time series models that were developed for ham and spam data respectively. First, an ARIMA (1,0,0) model was developed, but this was shown be inferior when the AIC scores were compared to the ARIMA(2,0,1) model. The Box-Ljung statistic (Figure 23) also demonstrated that the model was inadequate.

For the ham data, based on the Ljung Box Statistic that was applied to the ARIMA(2,0,1) model, the model is significant in this case [6], for the time series. The time series auto ARIMA was applied to the first difference of the ham time series, to obtain this model (Figures 24 – 26). The AIC is 2243.33. Based on the AIC value, the ARIMA model (3,0,2) is not as good as the auto-ARIMA function. From this analysis, the better model for ham here is ARMA (2,0,1).

A similar process was repeated for the spam data set, and the optimal time series was an ARIMA(1,0,1) time series (Figure 36), when compared to an ARIMA(3,0,2) time series and a ARIMA(1,0,0) function (Figures 34 - 35). Thus, the null hypothesis in both cases was rejected, since there was a time series function for both ham and spam.

3.3 Integrated Spam Filter Performance Results

At this point I have not performed testing on the integrated spam filter. Testing will occur with the acquisition of complete email message sets with ham and spam appropriately classified.

4 Recommendation

From the results obtained by this study, I recommend using a ARIMA(1,0,1) filter for the spam time series data and the ARIMA(2,0,1) filter for the ham time series data. When using a Generalized Linear Model (GLM), I recommend using a Stepwise Log-Transformed GLM-57 model to serve as a static filter.

Finally, when combining both these elements to develop a comprehensive spam filter, I recommend using Bayesian analysis to calculate the probability of the likelihood of the message being marked as ham or spam, respectively.

5 References

[1] Garfinkel, S., & Schwartz, A. (1998). *Chapter 1 what's spam and what's the problem?*. Retrieved November 5, 201fromhttp://oreilly.com/catalog/spam/chapter/ch01.html

[2] Edwards, J. (2007). *The real cost of spam.* Retrieved November 5, 2011, from http://www.itsecurity.com/features/real-cost-of-spam-121007/

[3] Edwards, J. (2008). *The success and failure of spam control.* Retrieved November 5, 2011, from http://www.networksecurityjournal.com/features/success-failure-spam-control-032608/

[4] D. E. Brown and X. Wang, “Laboratory 2: Spam Filters," October 2011, assignment in class SYS 4021.

[5] Spam E-mail Database , June- July 1999, Created by Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt, Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304

[6] D. E. Brown and X. Wang, “Laboratory 2: Spam Filters Template," October 2011, SYS 4021.

6 Appendices

6.1 Appendix A – Data & Images

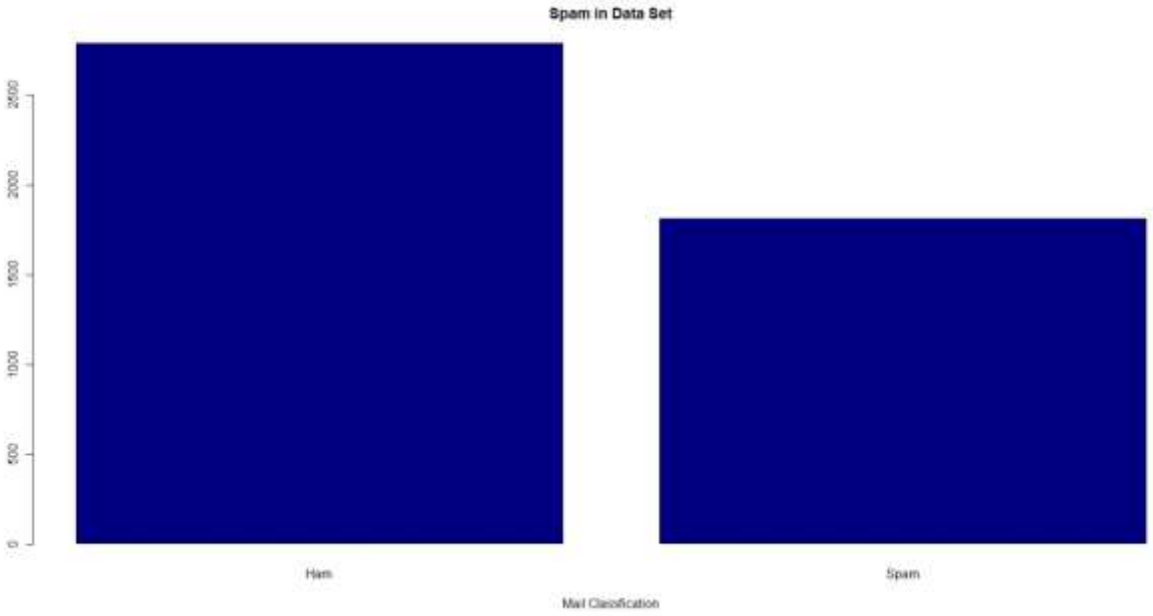

Figure 1 - Bar Chart for Spam and Ham

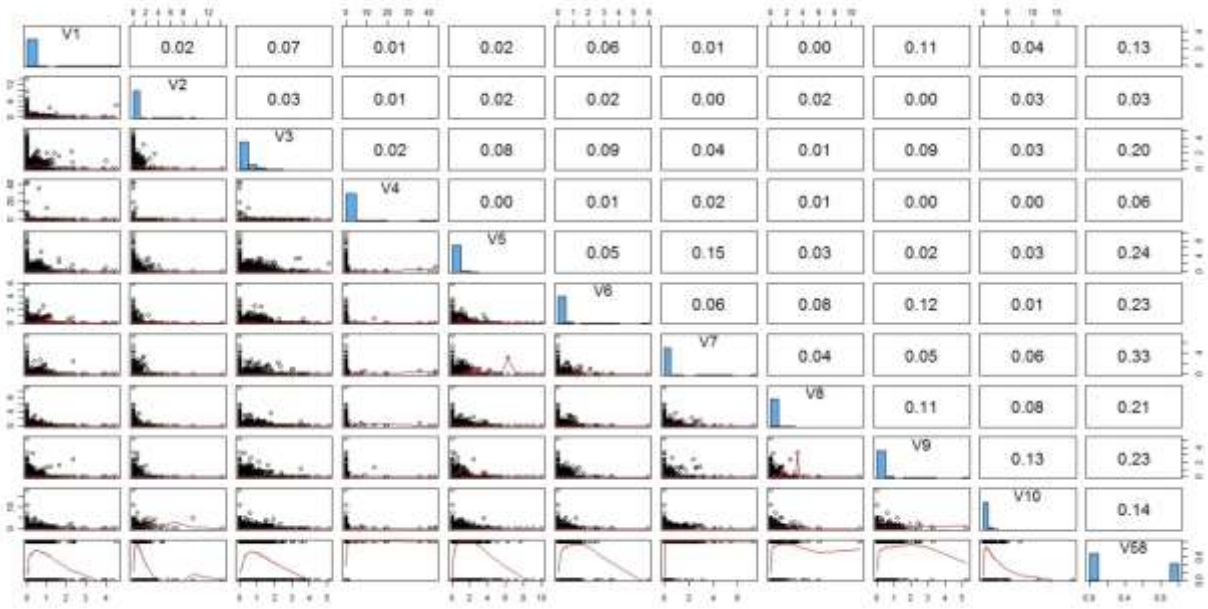

Figure 2 - Scatter Plot Matrix for Variables 1 – 10

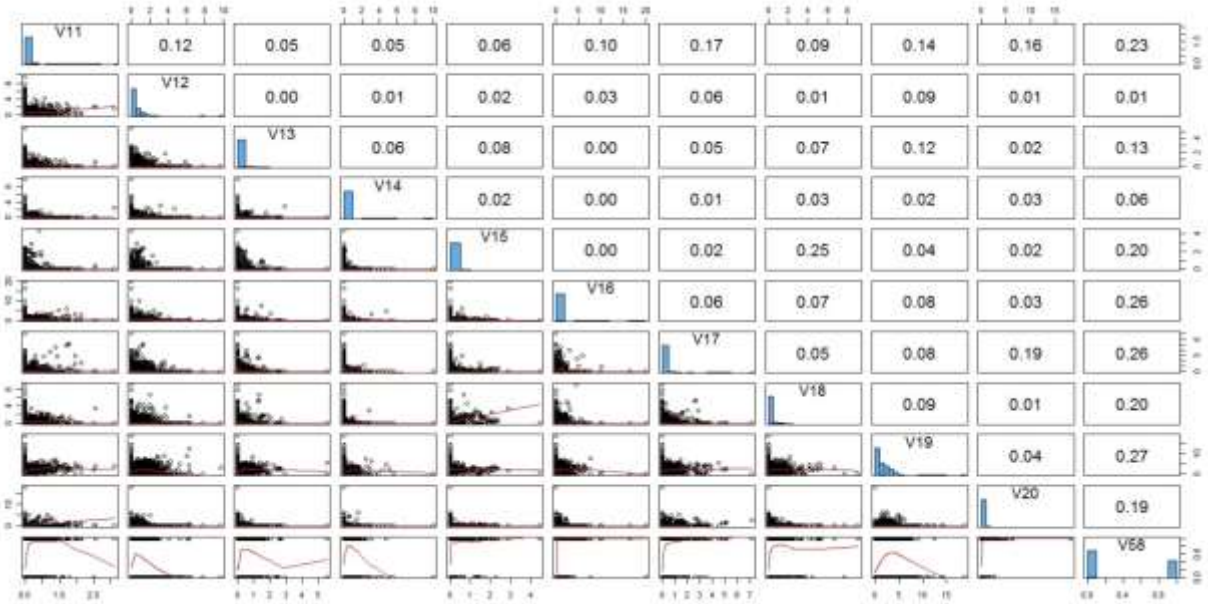

Figure 3 - Scatter Plot Matrix for Variables 11 - 20

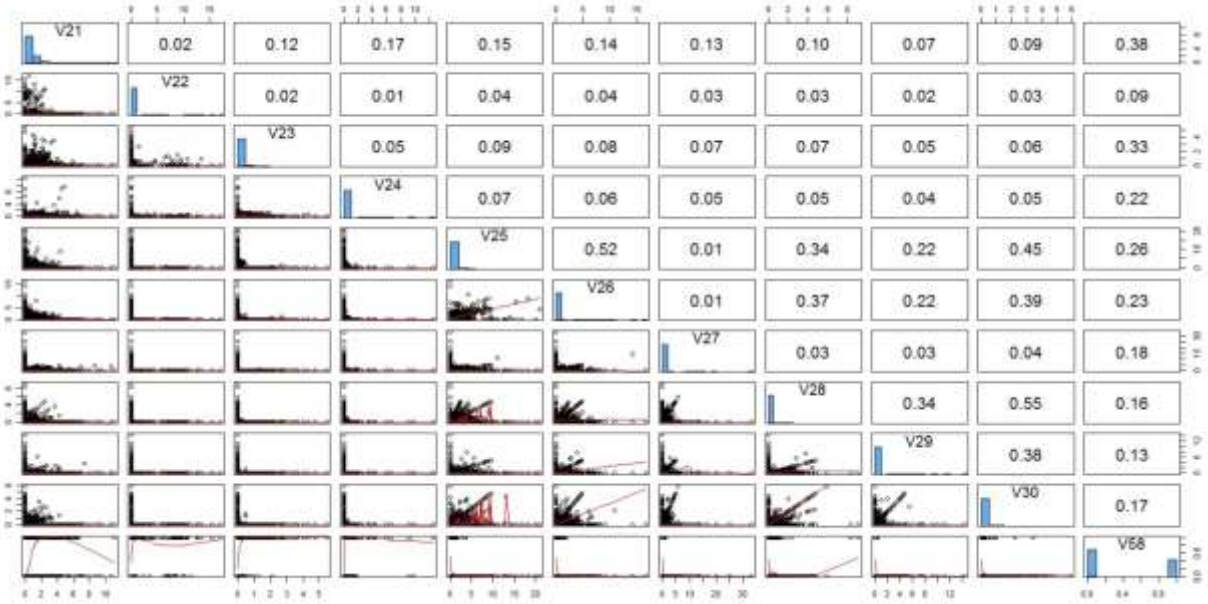

Figure 4 - Scatter Plot Matrix for Variables 21 - 30

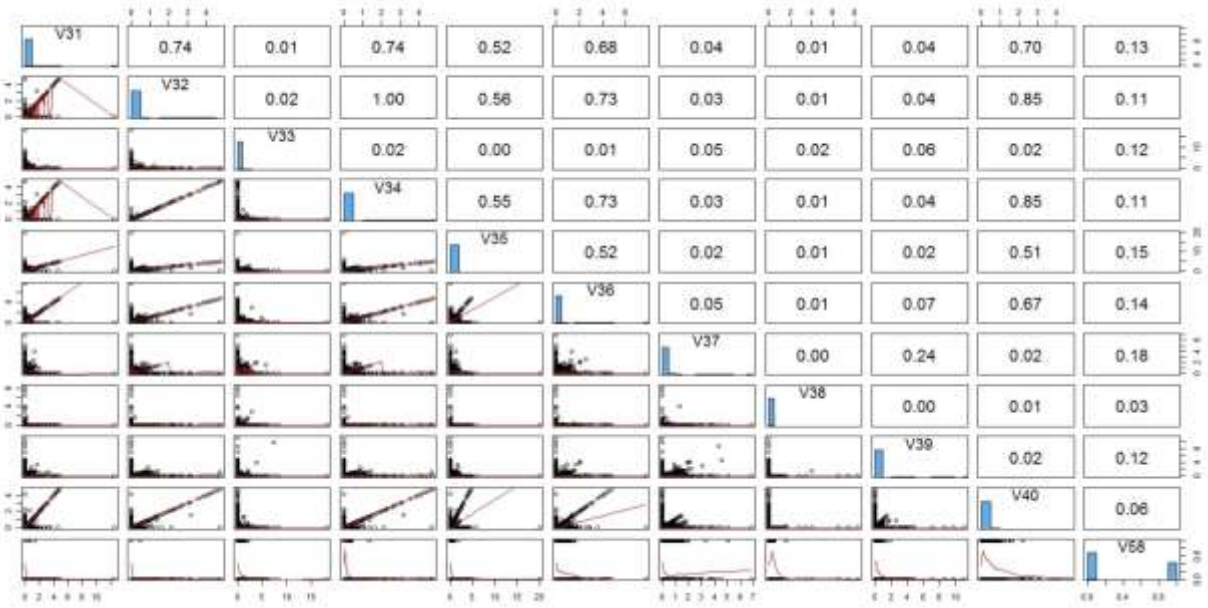

Figure 5 - Scatter Plot Matrix for Variables 31 - 40

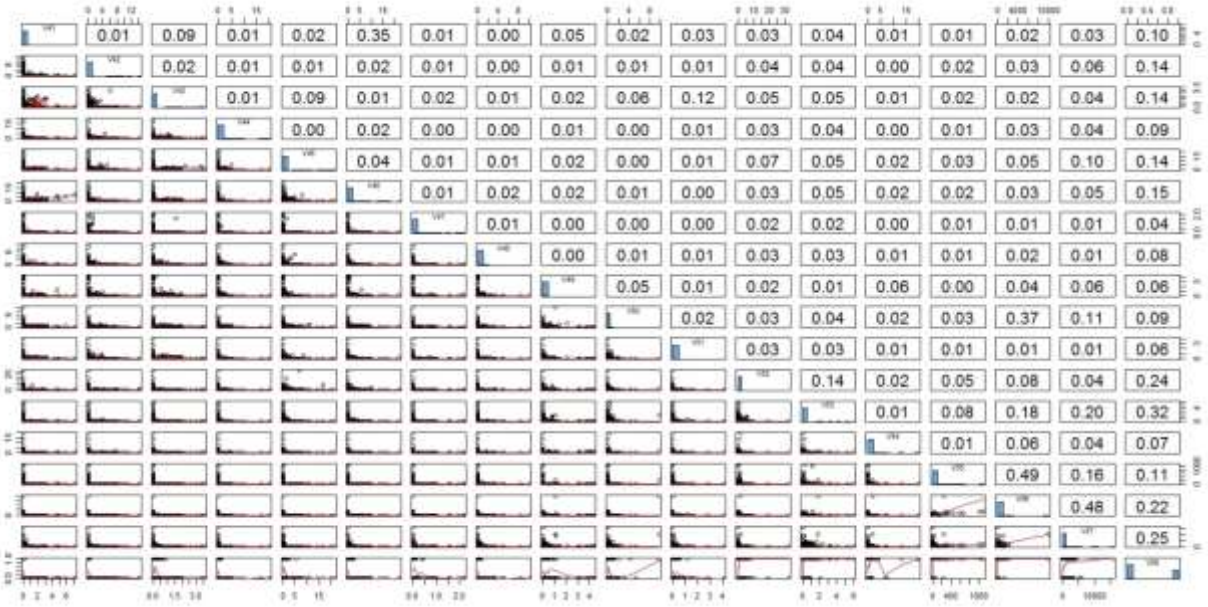

Figure 6 - Scatter Plot Matrix for Variables 41 - 57

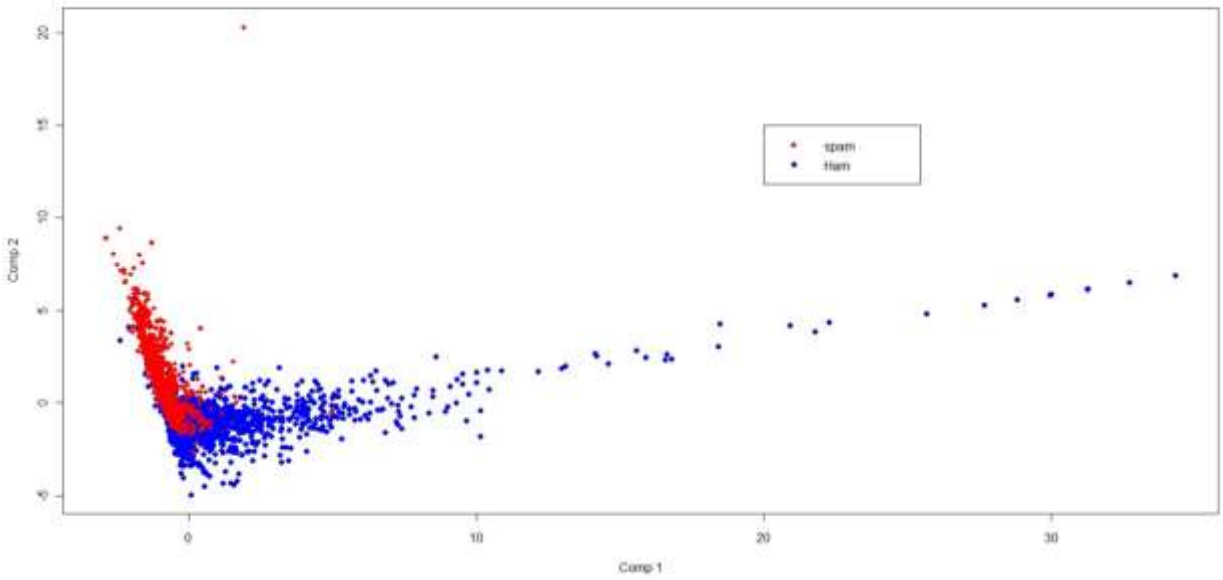

Figure 7 – PCA Color Graph

```
> anova(pc.n, spampca1, test = "Chi")
Analysis of Deviance Table

Model 1: r ~ 1
Model 2: r ~ Comp.1 + Comp.2 + Comp.3 + Comp.4 + Comp.5 + Comp.6 + Comp.7 +
    Comp.8 + Comp.9 + Comp.10 + Comp.11 + Comp.12 + Comp.13 +
    Comp.14 + Comp.15 + Comp.16 + Comp.17 + Comp.18 + Comp.19 +
    Comp.20 + Comp.21 + Comp.22 + Comp.23 + Comp.24 + Comp.25 +
    Comp.26 + Comp.27 + Comp.28 + Comp.29 + Comp.30 + Comp.31 +
    Comp.32 + Comp.33 + Comp.34 + Comp.35 + Comp.36 + Comp.37 +
    Comp.38 + Comp.39 + Comp.40 + Comp.41 + Comp.42 + Comp.43 +
    Comp.44 + Comp.45 + Comp.46 + Comp.47 + Comp.48 + Comp.49 +
    Comp.50 + Comp.51 + Comp.52
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      3082     4135.4
2      3030     1190.8 52   2944.6 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 8 - Principal Components Regression Results

```
Call:
glm(formula = response ~ V1 + V2 + V4 + V5 + V6 + V7 + V8 + V9 +
    V10 + V12 + V15 + V16 + V17 + V19 + V20 + V21 + V22 + V23 +
    V24 + V25 + V26 + V27 + V28 + V29 + V33 + V35 + V36 + V38 +
    V39 + V41 + V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 +
    V52 + V53 + V54 + V56 + V57, family = binomial, data = data.frame(spam[,
    -58], response = as.factor(spam[, 58])))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.2354  -0.1997   0.0000   0.1110   5.2484

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.552e+00  1.278e-01 -12.144  < 2e-16 ***
V1          -4.686e-01  2.156e-01  -2.173 0.029752 *
V2          -1.372e-01  6.541e-02  -2.098 0.035934 *
V4           2.257e+00  1.507e+00   1.497 0.134317
V5           5.656e-01  1.017e-01   5.560 2.70e-08 ***
V6           8.248e-01  2.446e-01   3.371 0.000748 ***
V7           2.261e+00  3.274e-01   6.906 4.99e-12 ***
V8           5.645e-01  1.663e-01   3.395 0.000687 ***
V9           6.683e-01  2.750e-01   2.430 0.015112 *
V10          1.161e-01  6.997e-02   1.659 0.097034 .
V12         -1.357e-01  7.331e-02  -1.851 0.064203 .
V15          1.293e+00  7.025e-01   1.841 0.065610 .
V16          1.048e+00  1.446e-01   7.250 4.16e-13 ***
V17          9.452e-01  2.208e-01   4.280 1.87e-05 ***
V19          8.972e-02  3.437e-02   2.610 0.009051 **
V20          1.117e+00  5.534e-01   2.018 0.043616 *
V21          2.330e-01  4.943e-02   4.714 2.43e-06 ***
V22          2.210e-01  1.648e-01   1.341 0.179849
V23          2.193e+00  4.674e-01   4.692 2.70e-06 ***
V24          4.424e-01  1.690e-01   2.618 0.008843 **
V25         -1.981e+00  3.130e-01  -6.329 2.47e-10 ***
V26         -1.036e+00  4.401e-01  -2.354 0.018558 *
V27         -1.122e+01  1.795e+00  -6.250 4.10e-10 ***
V28          4.182e-01  1.990e-01   2.102 0.035596 *
V29         -2.525e+00  1.525e+00  -1.656 0.097730 .
V33         -7.300e-01  3.081e-01  -2.370 0.017808 *
V35         -2.137e+00  7.833e-01  -2.729 0.006361 **
V36          9.643e-01  3.089e-01   3.121 0.001802 **
V38         -6.061e-01  4.274e-01  -1.418 0.156156
V39         -8.670e-01  3.829e-01  -2.264 0.023546 *
V41         -4.420e+01  2.643e+01  -1.673 0.094420 .
V42         -2.690e+00  8.448e-01  -3.184 0.001452 **
V43         -1.274e+00  8.230e-01  -1.548 0.121648
```

```
V43         -1.274e+00  8.230e-01  -1.548 0.121648
V44         -1.619e+00  5.351e-01  -3.026 0.002478 **
V45         -7.956e-01  1.546e-01  -5.147 2.64e-07 ***
V46         -1.466e+00  2.680e-01  -5.470 4.51e-08 ***
V47         -2.356e+00  1.793e+00  -1.314 0.188816
V48         -4.033e+00  1.564e+00  -2.579 0.009916 **
V49         -1.309e+00  4.474e-01  -2.926 0.003431 **
V52          3.588e-01  9.054e-02   3.963 7.39e-05 ***
V53          5.481e+00  7.062e-01   7.762 8.36e-15 ***
V54          2.202e+00  1.073e+00   2.052 0.040156 *
V56          1.041e-02  1.783e-03   5.836 5.35e-09 ***
V57          8.049e-04  2.114e-04   3.808 0.000140 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6170.2  on 4600  degrees of freedom
Residual deviance: 1824.9  on 4557  degrees of freedom
AIC: 1912.9

Number of Fisher Scoring iterations: 13
```

Figure 9 - Stepwise GLM-57 Model Results

```
> anova(spam.null, spam.glm, test = "Chi")
Analysis of Deviance Table

Model 1: response ~ 1
Model 2: response ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 +
    V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 +
    V21 + V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 +
    V31 + V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 +
    V41 + V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 + V50 +
    V51 + V52 + V53 + V54 + V55 + V56 + V57
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      4600     6170.2
2      4543     1815.8 57   4354.4 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> anova(spam.step, spam.glm, test = "Chi")
Analysis of Deviance Table

Model 1: response ~ V1 + V2 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V12 +
    V15 + V16 + V17 + V19 + V20 + V21 + V22 + V23 + V24 + V25 +
    V26 + V27 + V28 + V29 + V33 + V35 + V36 + V38 + V39 + V41 +
    V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 + V52 + V53 +
    V54 + V56 + V57
Model 2: response ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 +
    V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 +
    V21 + V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 +
    V31 + V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 +
    V41 + V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 + V50 +
    V51 + V52 + V53 + V54 + V55 + V56 + V57
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      4557     1824.9
2      4543     1815.8 14   9.1104    0.8239
> anova(spam.null, spam.step,test = "Chi")
Analysis of Deviance Table

Model 1: response ~ 1
Model 2: response ~ V1 + V2 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V12 +
    V15 + V16 + V17 + V19 + V20 + V21 + V22 + V23 + V24 + V25 +
    V26 + V27 + V28 + V29 + V33 + V35 + V36 + V38 + V39 + V41 +
    V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 + V52 + V53 +
    V54 + V56 + V57
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      4600     6170.2
2      4557     1824.9 43   4345.3 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 10 - Chi-Square Results for Model

Variable	Model
spam.null	Null Model of Spam
spam.glm	GLM-57
spam.step	Stepwise GLM-57

```
> anova(glm2.train, glm1.train, test = "Chi")
Analysis of Deviance Table

Model 1: response.train ~ V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 +
    V11 + V15 + V16 + V17 + V20 + V21 + V23 + V24 + V25 + V26 +
    V27 + V28 + V29 + V30 + V33 + V35 + V36 + V37 + V38 + V39 +
    V41 + V42 + V44 + V45 + V46 + V47 + V48 + V49 + V52 + V53 +
    V54 + V55 + V56 + V57
Model 2: response.train ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 +
    V10 + V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 +
    V20 + V21 + V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 +
    V30 + V31 + V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 +
    V40 + V41 + V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 +
    V50 + V51 + V52 + V53 + V54 + V55 + V56 + V57
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      3039     1182.4
2      3025     1170.9 14   11.477    0.6482
```

Figure 11 - Stepwise GLM Trained & Tested vs. GLM Trained & Tested

Variable	Model
Glm2.train	Step-wise GLM 57 Trained & Tested
Glm1.train	GLM-57 Trained and Tested

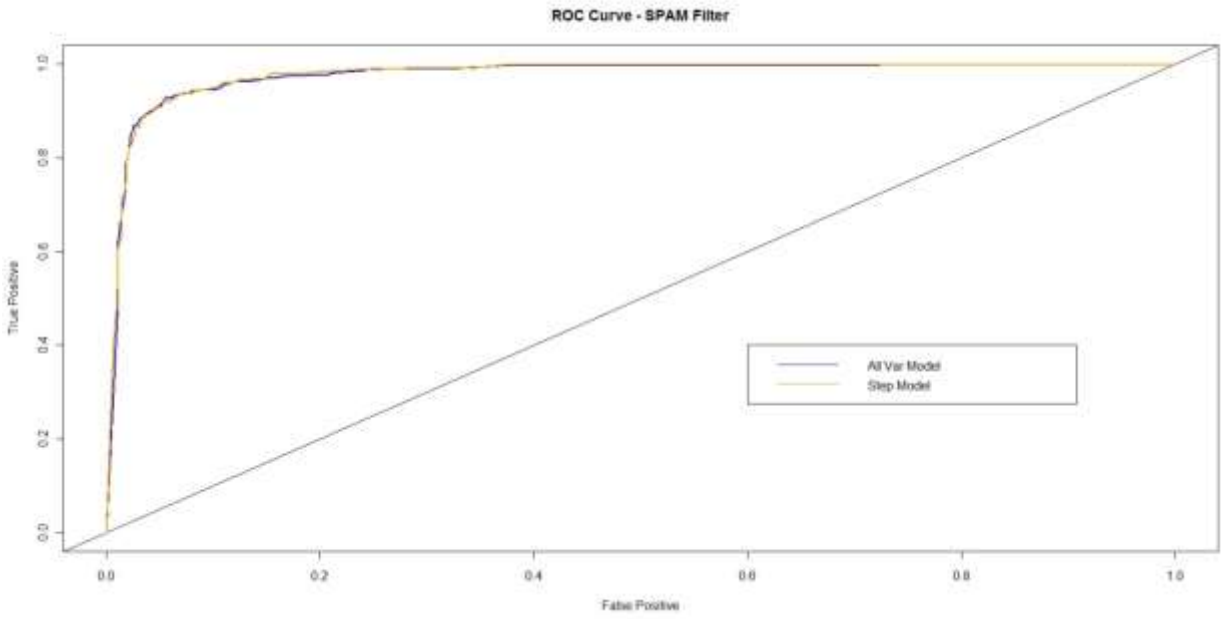

Figure 12 - Test Set GLM ROC Curves

```
> anova(Lspam2.train, Lspam1.train, test = "Chi")
Analysis of Deviance Table

Model 1: response.train ~ V3 + V5 + V7 + V8 + V11 + V12 + V13 + V14 +
    V16 + V17 + V18 + V20 + V21 + V23 + V24 + V25 + V27 + V28 +
    V33 + V35 + V37 + V38 + V41 + V42 + V43 + V44 + V45 + V46 +
    V48 + V51 + V52 + V53 + V54 + V55 + V57
Model 2: response.train ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 +
    V10 + V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 +
    V20 + V21 + V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 +
    V30 + V31 + V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 +
    V40 + V41 + V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 +
    V50 + V51 + V52 + V53 + V54 + V55 + V56 + V57
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      3047     933.47
2      3025     916.89 22   16.577    0.7863
```

Figure 13 - Chi-Square Comparisons for Log-Transform Models

Variable	Model
Lspam2.train	Step-wise Log-Transform GLM-57
Lspam1.train	Log Transform GLM-57

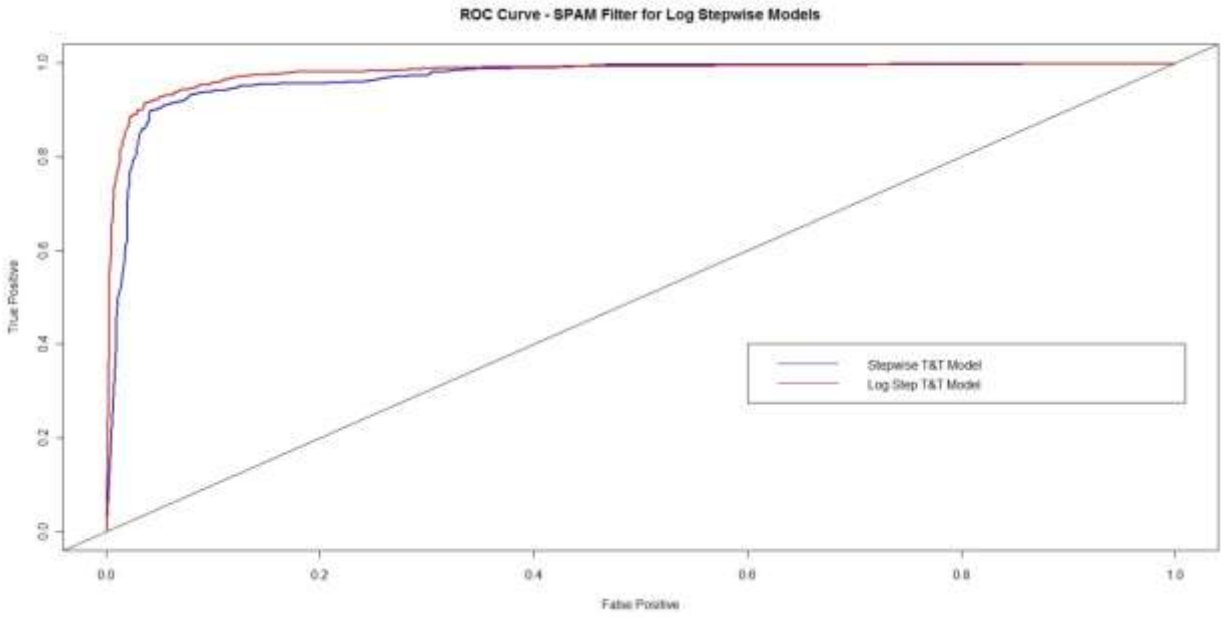

Figure 14 - ROC Curves for Stepwise Trained & Tested vs. Stepwise Log-Transformed Trained & Tested

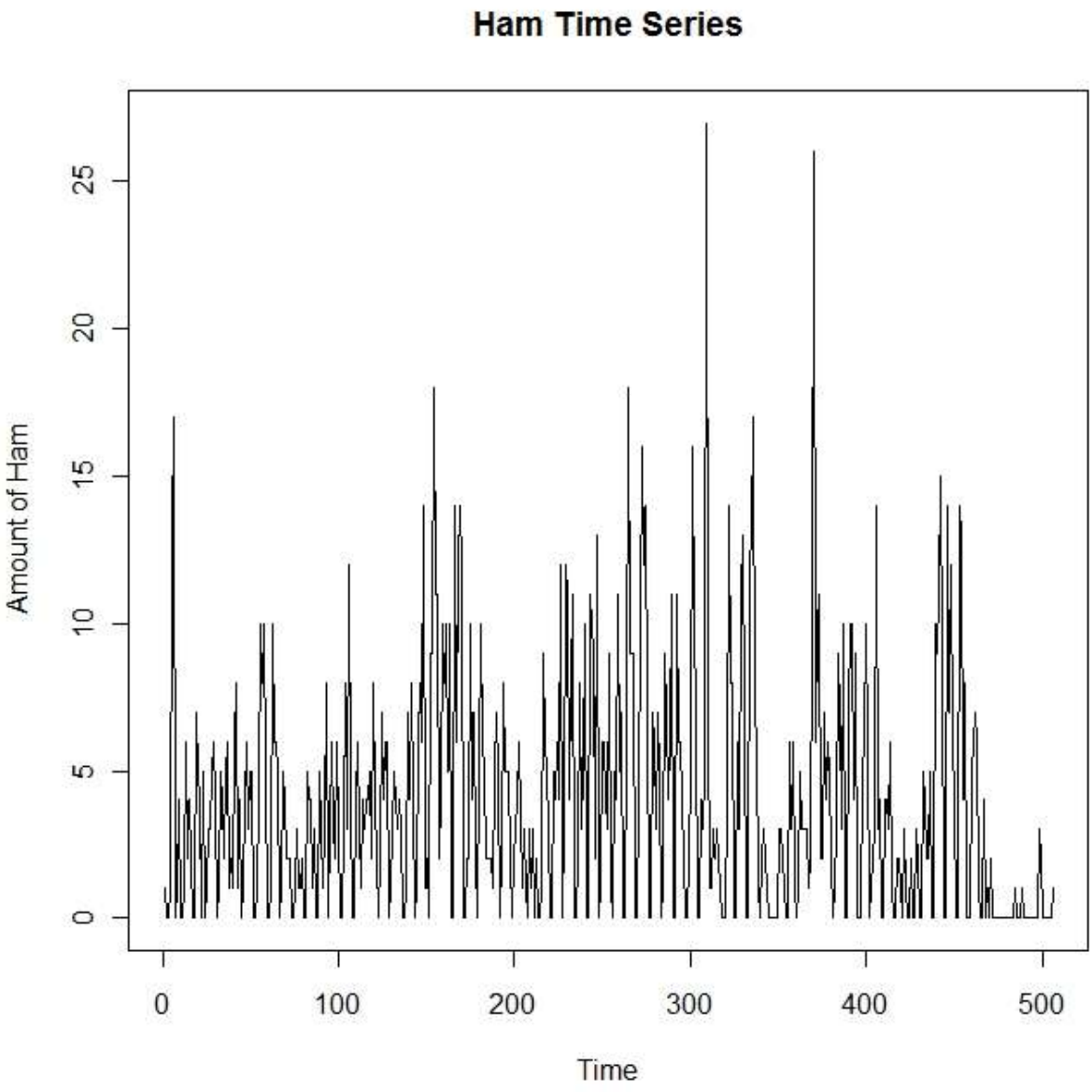

Figure 15 - Ham Time Series

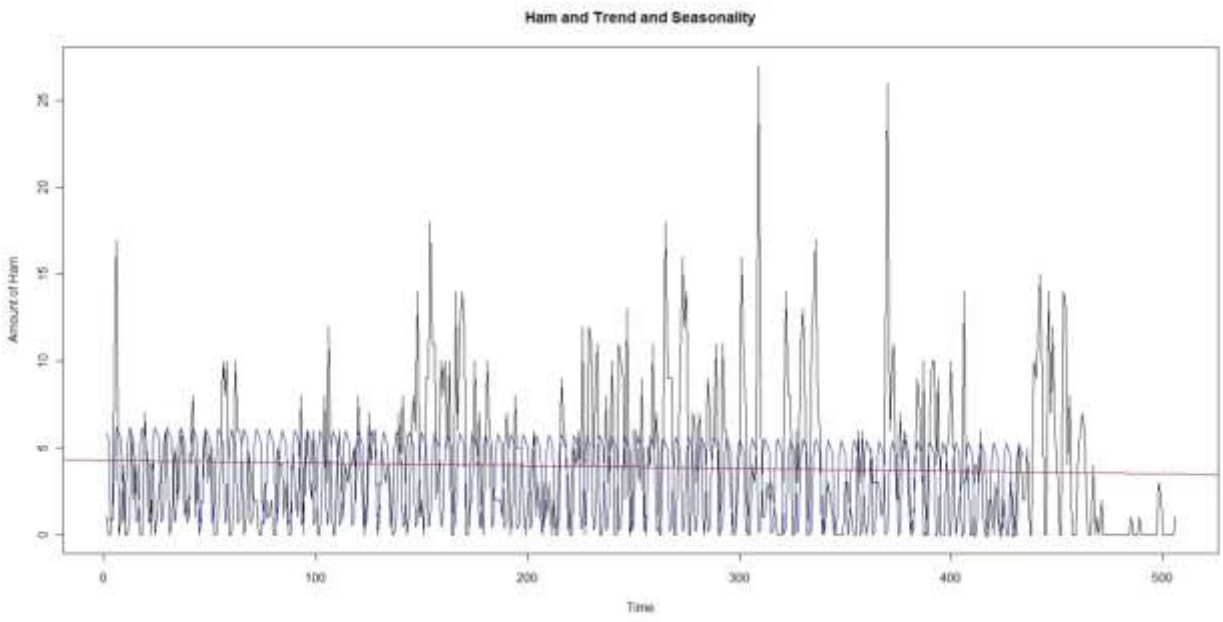

Figure 16 - Ham Trend and Seasonality

```
> summary(ham.trend)

Call:
lm(formula = ham.ts ~ time)

Residuals:
   Min     1Q Median     3Q    Max
-4.267 -3.533 -1.082  2.030 23.199

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.269589   0.374610  11.397   <2e-16 ***
time        -0.001516   0.001280  -1.184    0.237
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.207 on 504 degrees of freedom
Multiple R-squared: 0.002772,   Adjusted R-squared: 0.0007938
F-statistic: 1.401 on 1 and 504 DF,  p-value: 0.2371
```

Figure 17 - Ham Trend Significance Test

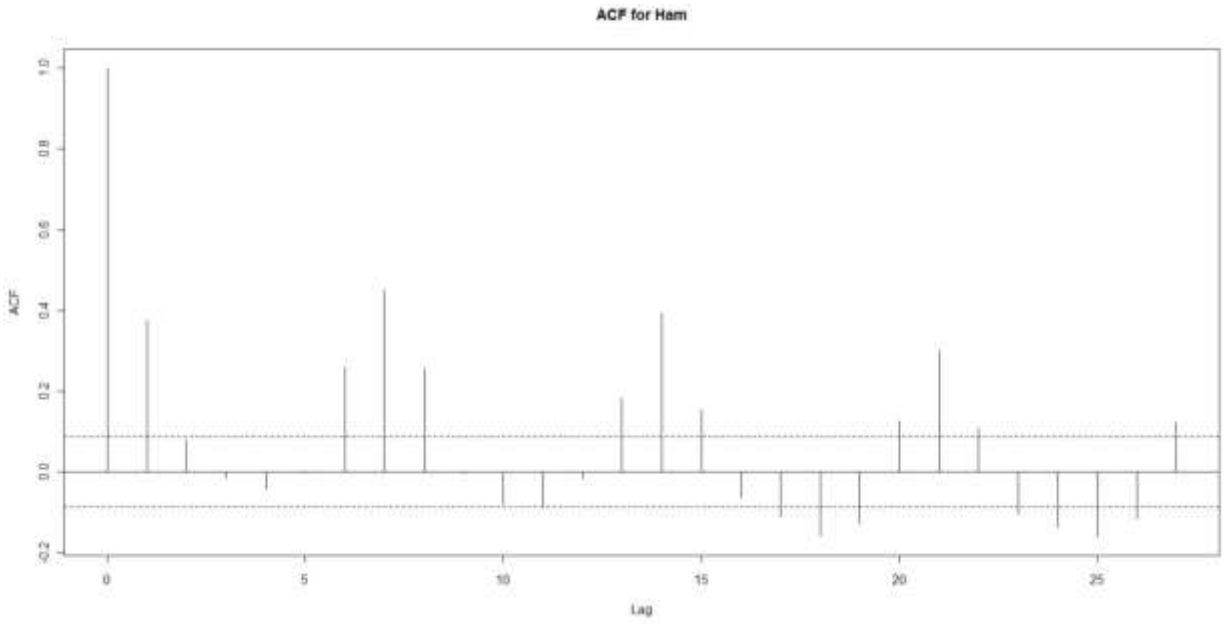

Figure 18 - Ham ACF Graph

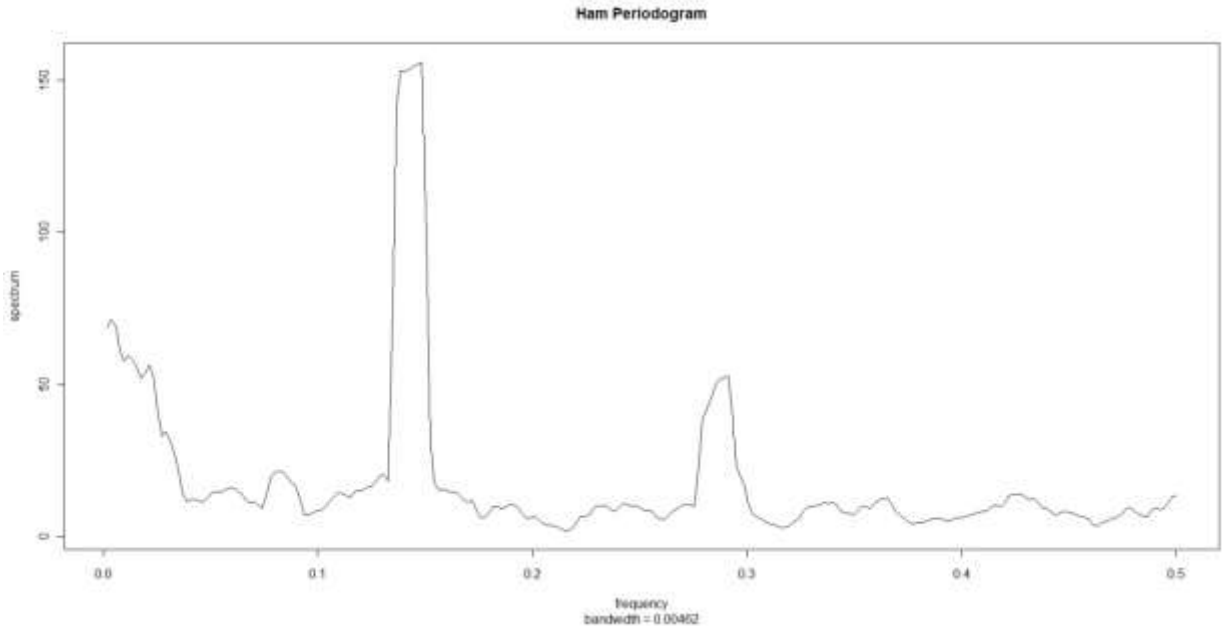

Figure 19 - Ham Periodogram

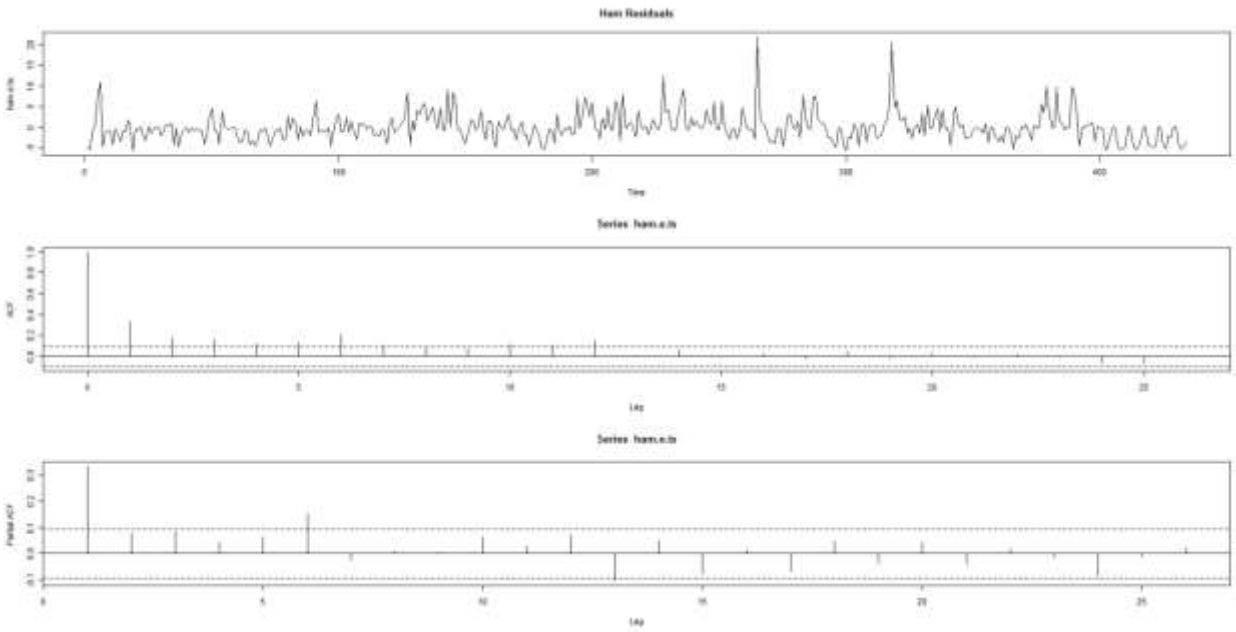

Figure 20 - Ham Residuals Pattern, ACF, PACF

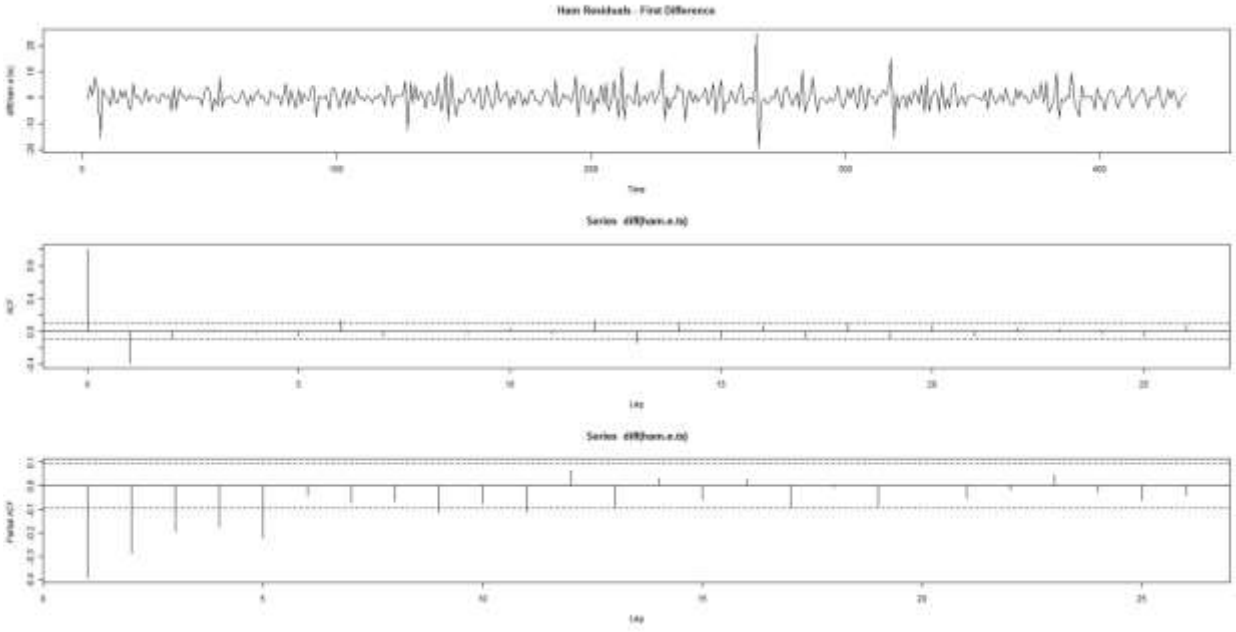

Figure 21 - Ham Residuals First Order Difference Pattern, ACF, PACF

```
> ham.ar1 <- arima(diff(ham.e.ts),order=c(1,0,0))
> ham.ar1

Call:
arima(x = diff(ham.e.ts), order = c(1, 0, 0))

Coefficients:
          ar1  intercept
      -0.3887     0.0027
s.e.   0.0442     0.1252

sigma^2 estimated as 13.07:  log likelihood = -1170.92,  aic = 2347.84
```

Figure 22 - HAM ARIMA(1,0,0) Model Results

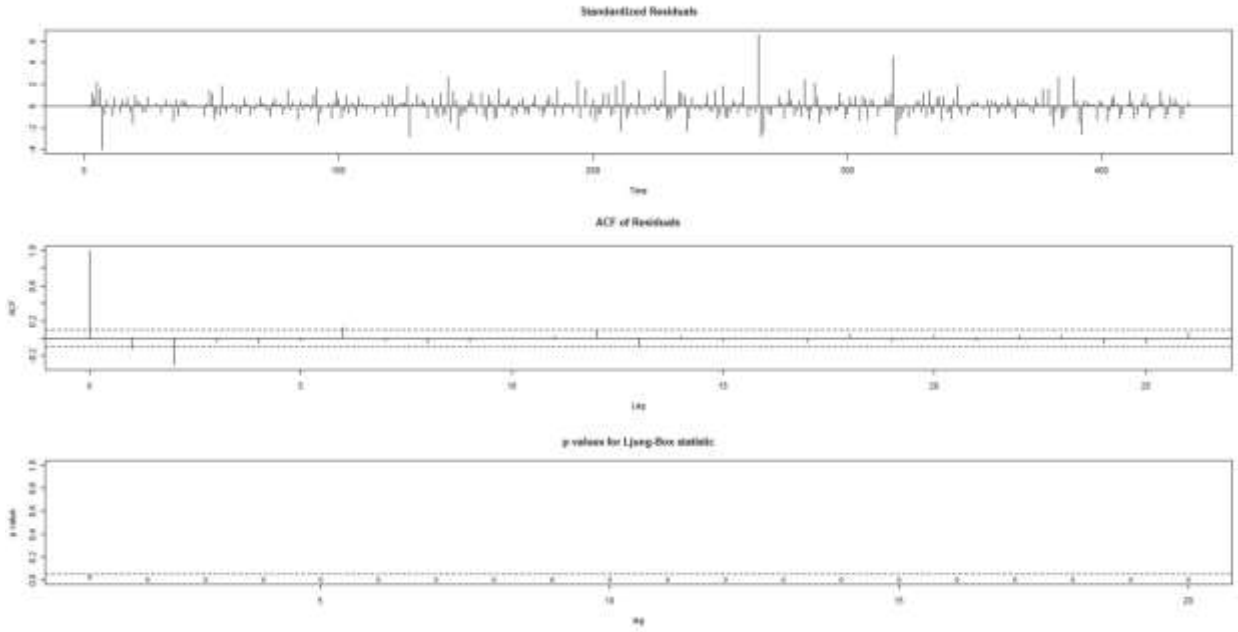

Figure 23 - Ham ARIMA(1,0,0) ACF and Box-Ljung Graph

```
> ham.auto
Series: diff(ham.e.ts)
ARIMA(2,0,1) with zero mean

Call: auto.arima(x = diff(ham.e.ts))

Coefficients:
         ar1     ar2      ma1
      0.2556  0.0341  -0.9542
s.e.  0.0600  0.0583   0.0361

sigma^2 estimated as 10.18:  log likelihood = -1117.67
AIC = 2243.33   AICc = 2243.42   BIC = 2259.61
```

Figure 24 - Ham Auto-ARIMA(2,0,1) Model Results

```
> ham.arma.32 <- arima(diff(ham.e.ts),order=c(3,0,2))
> ham.arma.32
Series: diff(ham.e.ts)
ARIMA(3,0,2) with non-zero mean

Call: arima(x = diff(ham.e.ts), order = c(3, 0, 2))

Coefficients:
         ar1      ar2     ar3      ma1     ma2  intercept
      1.0541  -0.1796  0.0321  -1.7661  0.7687    -0.0009
s.e.     NaN   0.0627  0.0519      NaN     NaN     0.0046

sigma^2 estimated as 10.05:  log likelihood = -1115.26
AIC = 2244.52   AICc = 2244.79   BIC = 2273.02
```

Figure 25 - Ham ARIMA(3,0,2) Model Results

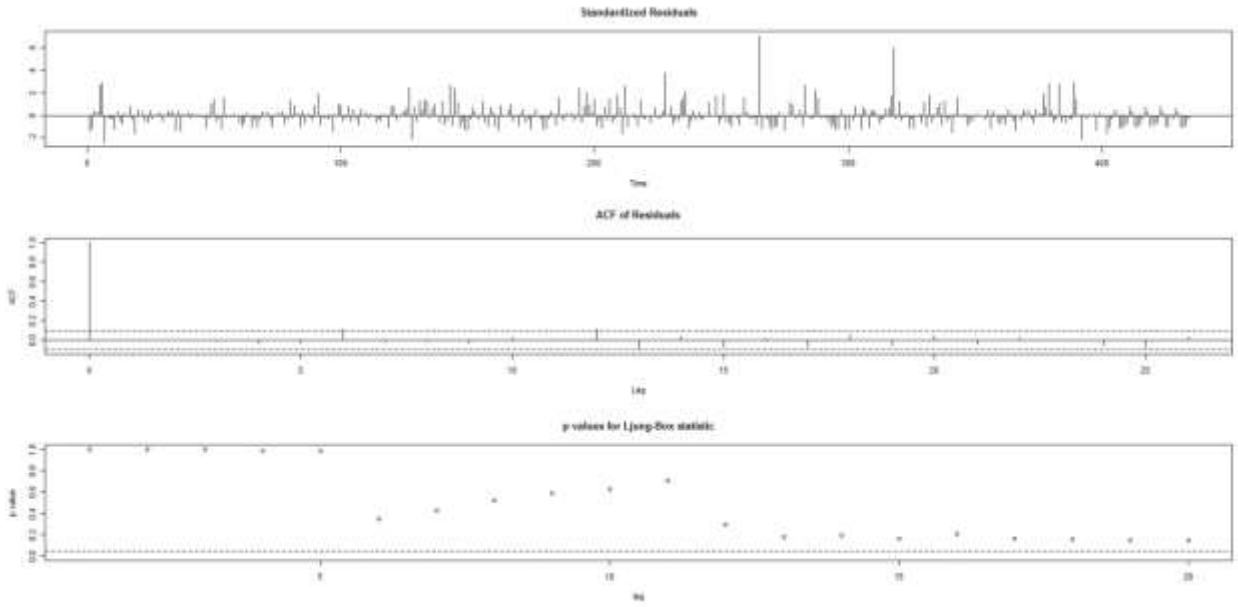

Figure 26 - Ham ARIMA(2,0,1) ACF & Box-Ljung Graph

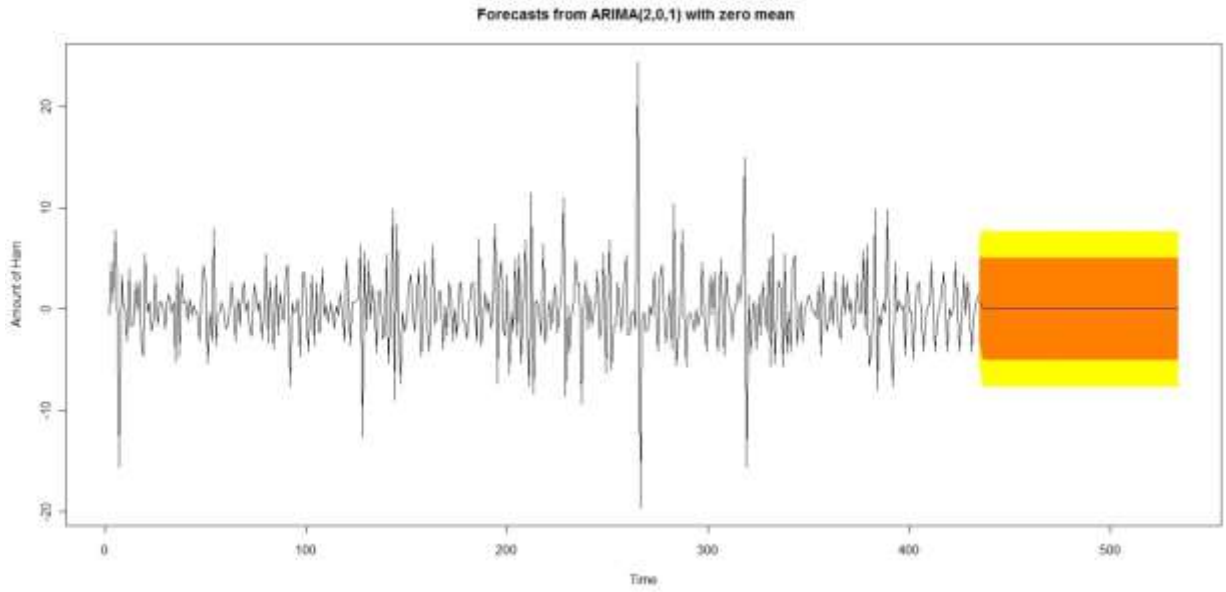

Figure 27 - Ham Forecasted Model for 100 time units in the future

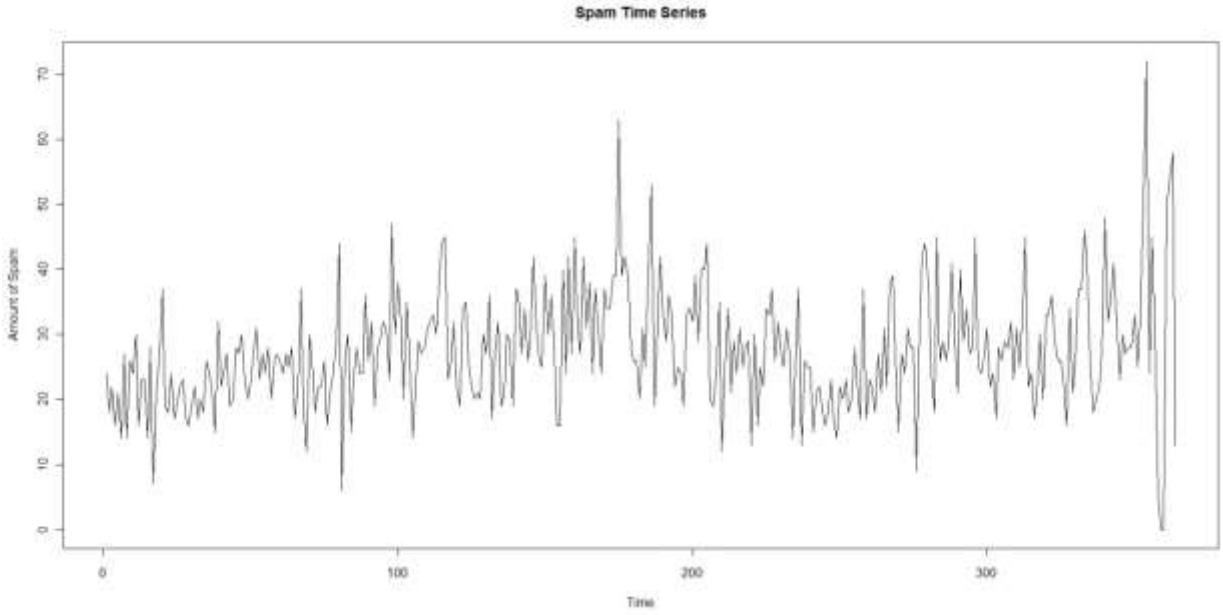

Figure 28 - Spam Time Series Graph

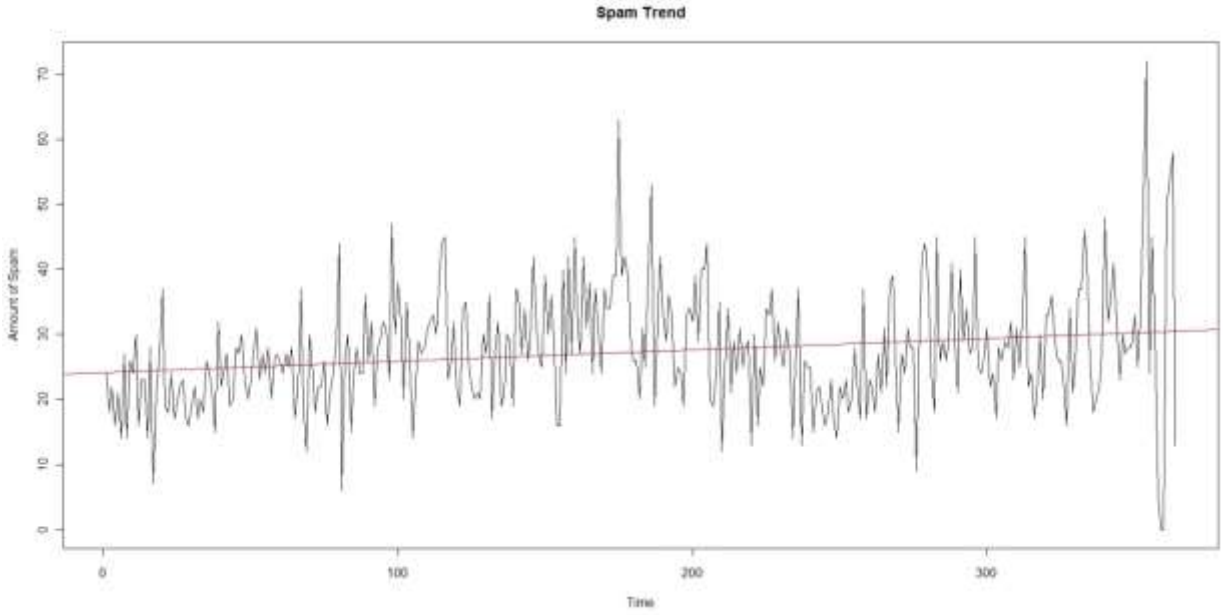

Figure 29 - Spam Trend Graph

```
> summary(spam.trend)

Call:
lm(formula = spam.ts ~ time2)

Residuals:
    Min      1Q  Median      3Q     Max
-30.479  -5.633  -0.786   4.623  41.626

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 24.175415   0.940017  25.718  < 2e-16 ***
time2        0.017509   0.004464   3.923 0.000105 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.949 on 362 degrees of freedom
Multiple R-squared: 0.04077,    Adjusted R-squared: 0.03812
F-statistic: 15.39 on 1 and 362 DF,  p-value: 0.0001048
```

Figure 30 - Spam Trend Results

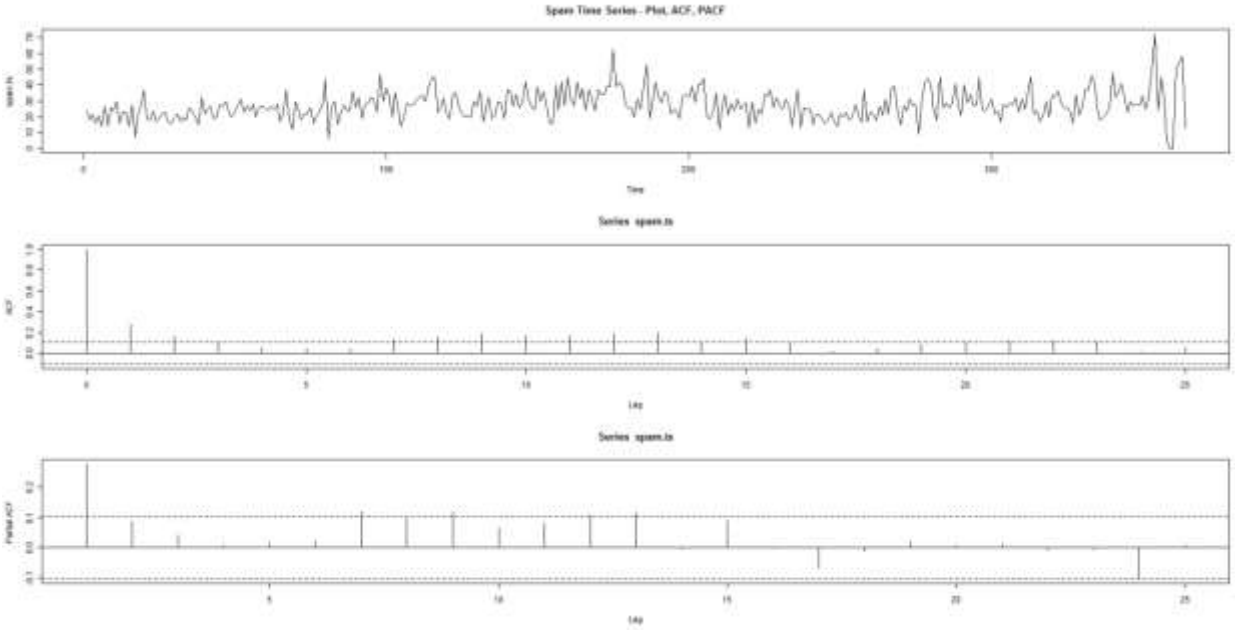

Figure 31 - Spam Time Series, ACF, PACF Plots

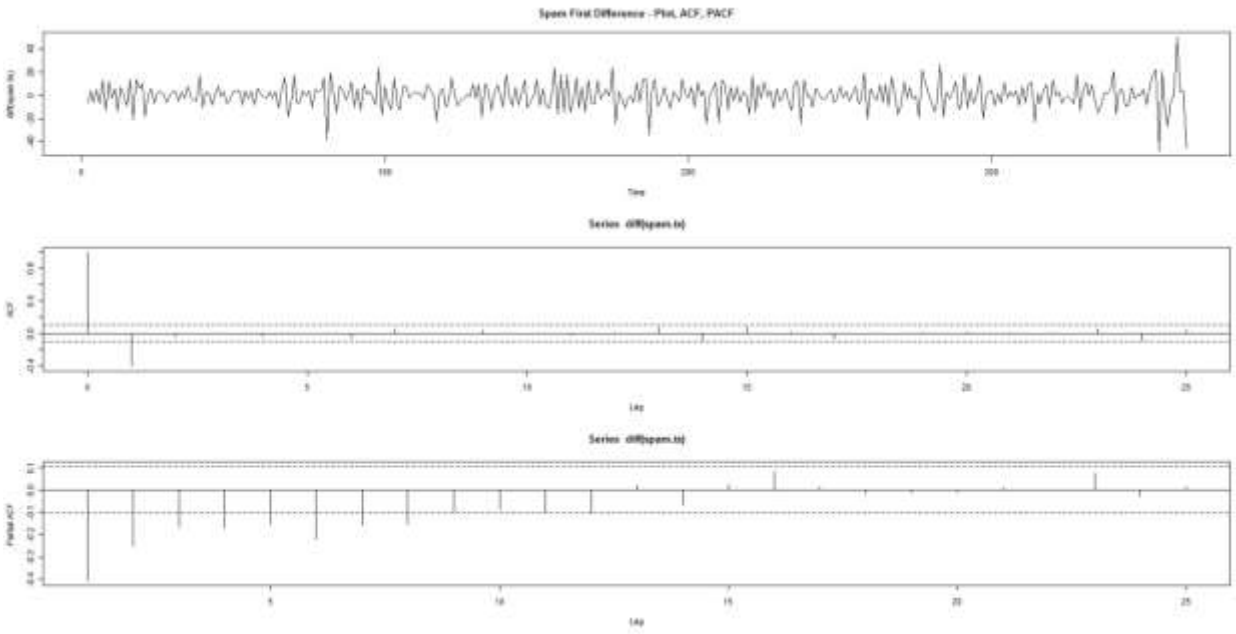

Figure 32 - Spam First Order Difference Plot, ACF, PACF

```
> spam.ar1
Series: spam.ts
ARIMA(1,0,0) with non-zero mean

Call: arima(x = spam.ts, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.2759    27.3523
s.e.  0.0505     0.6333

sigma^2 estimated as 76.72:  log likelihood = -1306.44
AIC = 2618.87   AICc = 2618.94   BIC = 2630.56
```

Figure 33 - Spam ARIMA(1,0,0) Model Results

```
> spam.auto
Series: diff(spam.ts)
ARIMA(1,0,1) with zero mean

Call: auto.arima(x = diff(spam.ts))

Coefficients:
         ar1      ma1
      0.161  -0.9449
s.e.  0.056   0.0180

sigma^2 estimated as 73.93:  log likelihood = -1297.05
AIC = 2600.11   AICc = 2600.17   BIC = 2611.79
```

Figure 34 - Spam Auto-ARIMA(1,0,1)

```
> spam.arma.32 <- arima(diff(spam.ts),order=c(3,0,2))
> spam.arma.32

Call:
arima(x = diff(spam.ts), order = c(3, 0, 2))

Coefficients:
          ar1     ar2     ar3     ma1      ma2  intercept
      -0.8210  0.1609  0.0179  0.0497  -0.9503     0.0288
s.e.   0.0561  0.0763  0.0565  0.0221   0.0218     0.0286

sigma^2 estimated as 72.52:  log likelihood = -1294.55,  aic = 2603.11
```

Figure 35 - Spam ARIMA(3,0,2) Model Results

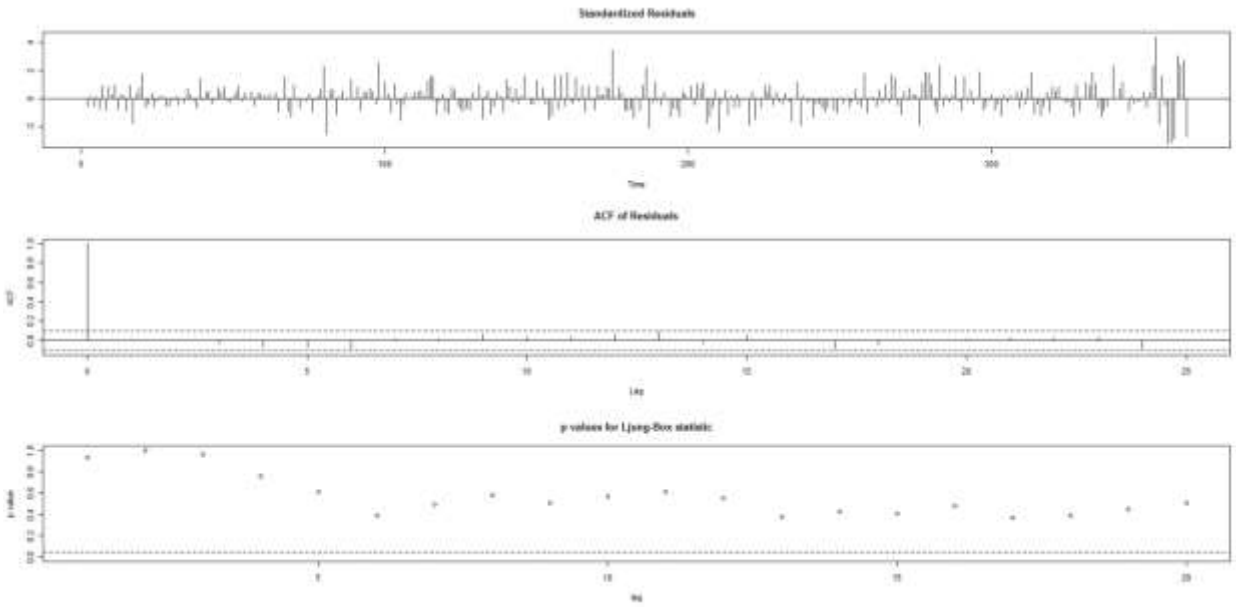

Figure 36 - Spam ARIMA(1,0,1) ACF & Box-Ljung Statistic

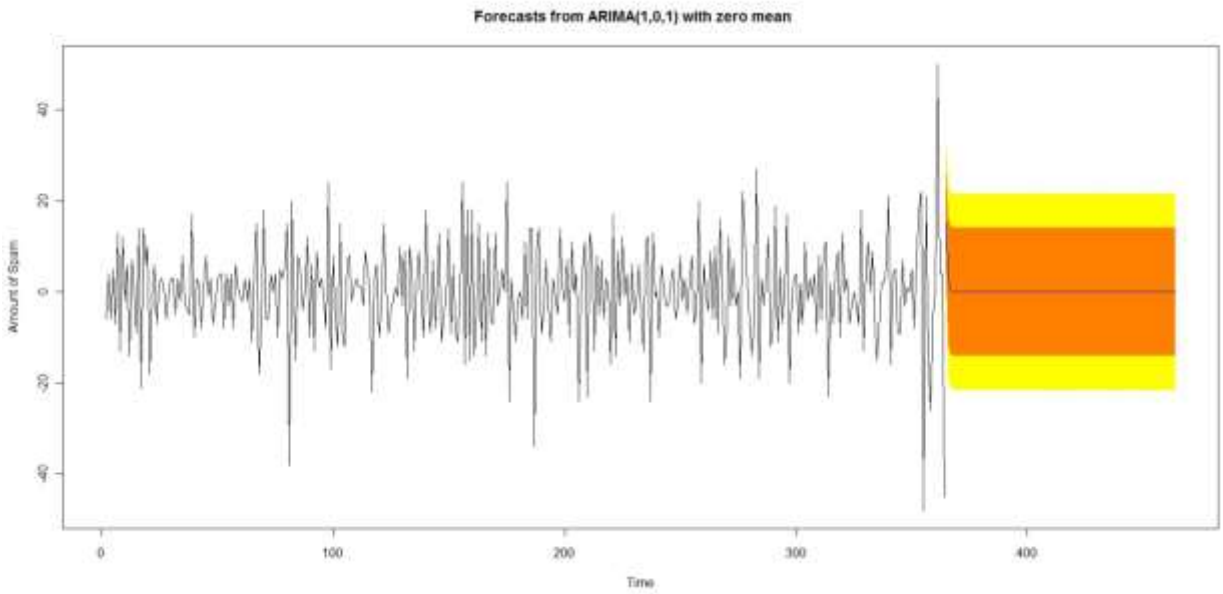

Figure 37 - Spam ARIMA(1,0,1) Forecast for 100 time units in the future