Project 2b: Time Series Spam Filter

Shareen Arshad, Chandler Dalton, Fan Feng

November 19, 2017

R Markdown

1. Ham Series Series Analysis: Build a time series filter for ham e-mail.

a) Perform a graphical analysis of ham.ts- use time series plots, periodograms, autocorrelation, and partial autocorrelation plots.

```
#Import Ham Dataset, change paths to run

ham<-read.table('C:/Users/student/Desktop/Train
Data/ham_ts.csv',header=T,sep=',')

#Displays time series data - sequence of data that have been observed in
successive order at different points in time

#Summarize the new ham data set

summary(ham)
```

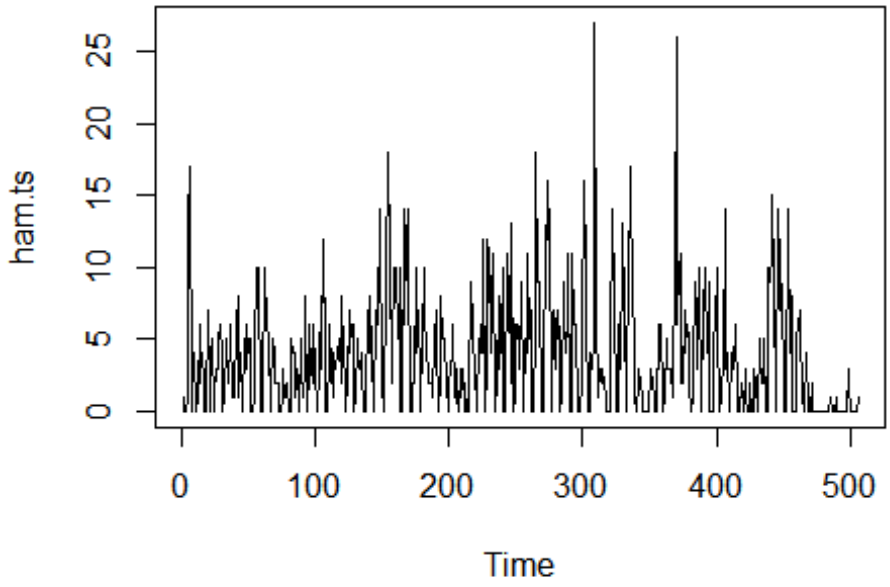
```
##       year          month             day            count
##  Min.   :2000   Min.   : 1.000   Min.   : 1.00   Min.   : 0.000
##  1st Qu.:2000   1st Qu.: 3.000   1st Qu.: 8.00   1st Qu.: 0.000
##  Median :2000   Median : 5.000   Median :16.00   Median : 3.000
##  Mean   :2000   Mean   : 5.599   Mean   :15.91   Mean   : 3.885
##  3rd Qu.:2001   3rd Qu.: 8.000   3rd Qu.:23.00   3rd Qu.: 6.000
##  Max.   :2001   Max.   :12.000   Max.   :31.00   Max.   :27.000
```

```
#Interpretation: Represents time series data; variables include year, month,
day, and count; Ranges for Year: (2000, 2001), Month: (1,12), Day: (1:31),
Count(0, 27)

#The ts() command can be used to get a time series of ham amount using ham
data from all years
ham.ts<-ts(ham$count)

#Load the forecast library, make sure to have package installed by
'install.packages("forecast")'
library(forecast)
```

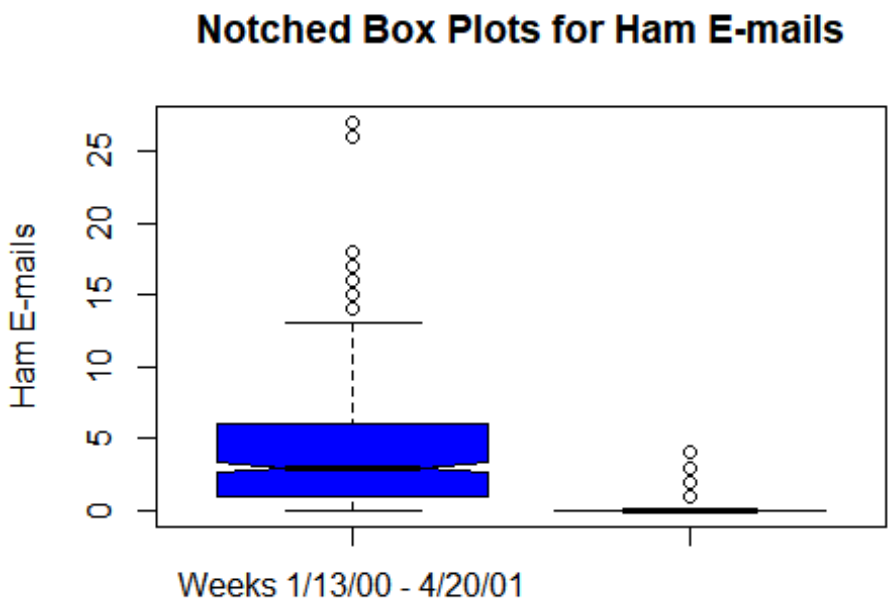
```
## Warning: package 'forecast' was built under R version 3.4.2
```

```
#TIME SERIES PLOTS
```

```r
#First, we must plot the ham time series created from the data
plot(ham.ts)
```

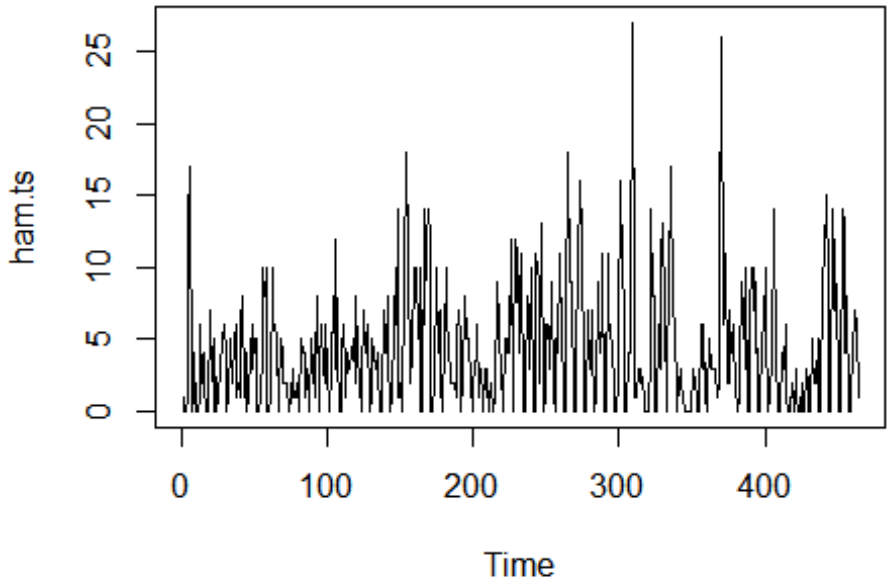



```r
#Interpretation: The daily time series for ham.ts shows a change in sampling
during the last six weeks.

#We can view if there is a change in sampling by viewing the boxplots for the
separate set of weeks, as

boxplot(ham.ts[1:464], ham.ts[465:506], notch = T, ylab = 'Ham E-mails',
names = c('Weeks 1/13/00 - 4/20/01', 'Weeks 4/20/01 - 6/1/01'), main =
'Notched Box Plots for Ham E-mails', col = c('blue', 'red'))
```

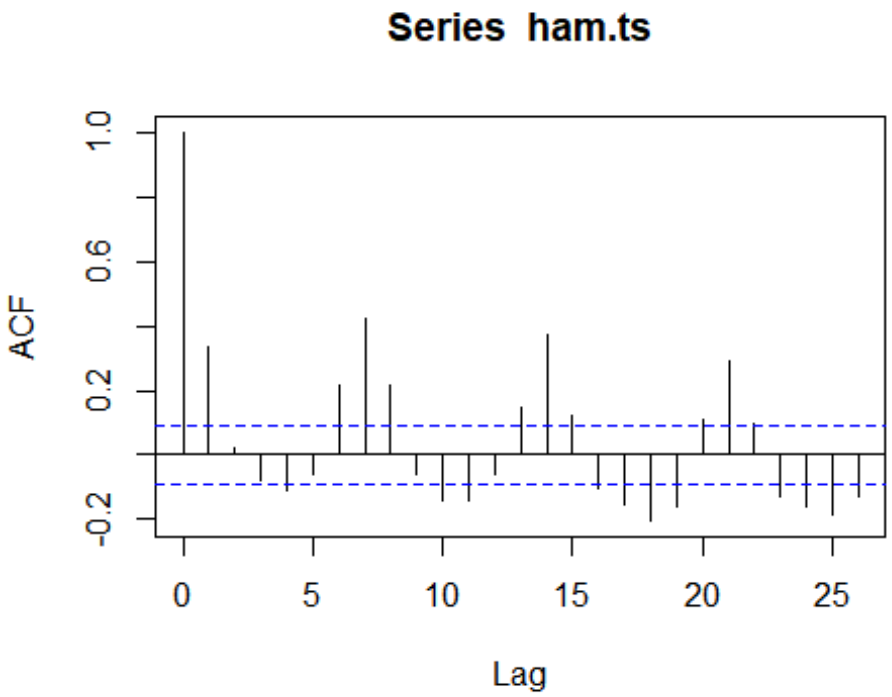



```
#Interpretation: From the Notched Box Plots, it is clear that the last 6
weeks of ham e-mails have a different sampling than the rest of the ham
emails. The median, max and min are 0 for the last six weeks, while the
median is at about 3 for the rest of the emails. Therefore, the six weeks can
skew the rest of the data.

#We must test for the change in sampling in the last six weeks by performing
a Wilcoxon test for the means, completed with the code as follows. This is
done on days 1-464 to days 465-506 (the last six weeks).
wilcox.test(ham.ts[1:464],ham.ts[465:506])

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  ham.ts[1:464] and ham.ts[465:506]
## W = 16270, p-value = 3.113e-13
## alternative hypothesis: true location shift is not equal to 0

#Due to the extremely low p-value of 3.113e-13, the null hypothesis can be
rejected that the means are equal and we can conclude that the last six weeks
have are a sampling error. Therefore, we can remove the last six weeks from
the data.

#Remove the last six weeks from ham.ts
ham.ts <-ts(ham$count[1:464])
```

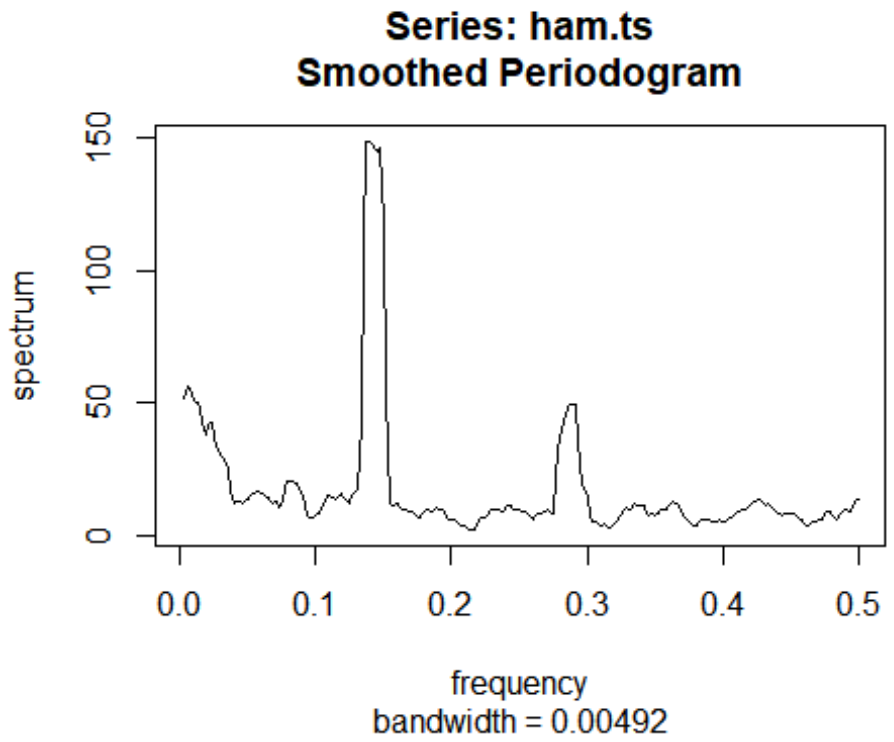
```r
#Plot the final ham time series
plot(ham.ts)
```




```r
#The ham.ts does not include the last 6 weeks. There is now no sampling error present in the graph.

#AUTOCORRELATION

#Use the acf() command to create the time series for ham.
acf(ham.ts)
```



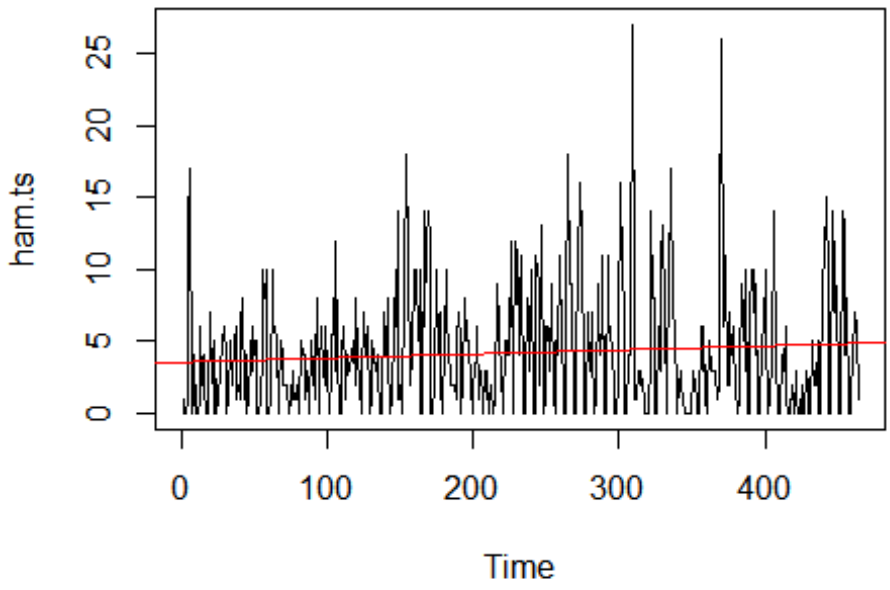

```
#There are significant autocorrelations in the ham time series at least until
lag = 25. There is a clear seasonality in the ham time series.

#PERIODIGRAM

#This can be further explored through a periodigram. The peak in the
frequency can be analyzed in order to find the period for the ham.ts.
pg.ham<-spec.pgram(ham.ts,spans=9,demean=T,log='no') #Periodigram
```




```
max.omega.ham<-pg.ham$freq[which(pg.ham$spec==max(pg.ham$spec))] #Find peak
1/max.omega.ham # 7.164 day period

## [1] 7.164179

# The period of hams is closely around 7 days, therefore a weekly period has
been identified.

#MODEL TREND OF HAM

#We must model the trend of ham to find if this is present in the data.

#Create a new variable time.ham which is a matrix of (length(ham.ts))
time.ham<-c(1:length(ham.ts))

#Build a new model, ham.trend which predicts ham.ts based on the time
variable, time.ham
ham.trend<-lm(ham.ts~time.ham)
summary(ham.trend)

##
## Call:
## lm(formula = ham.ts ~ time.ham)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7805 -3.5275 -0.8781  2.0615 22.6008
```

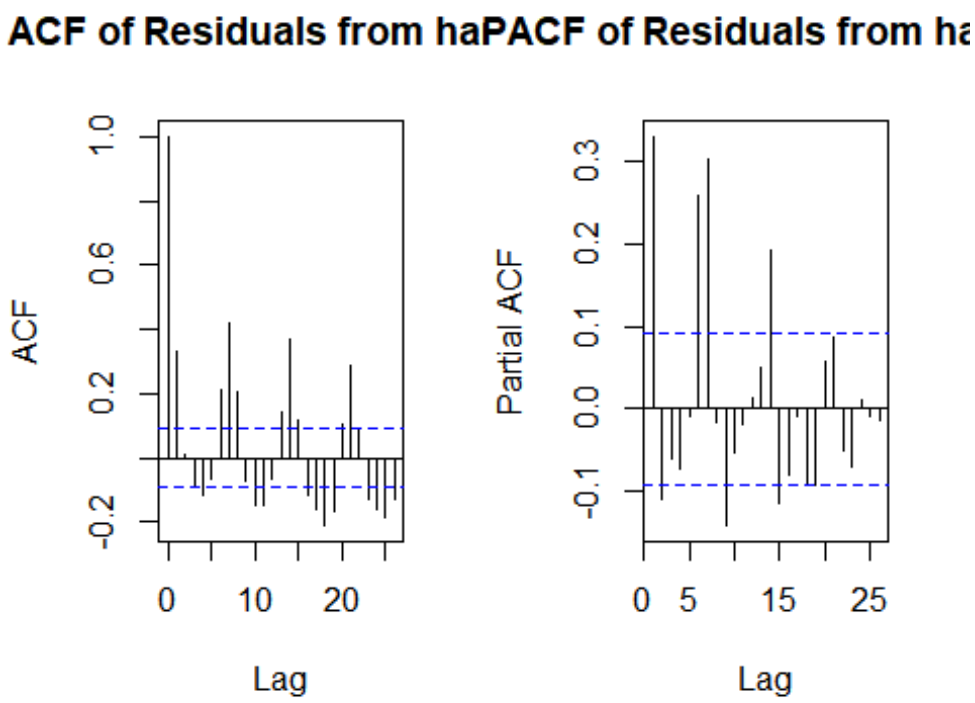
```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.613782   0.393934   9.174   <2e-16 ***
## time.ham    0.002542   0.001468   1.731   0.0841 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.236 on 462 degrees of freedom
## Multiple R-squared:  0.006446,   Adjusted R-squared:  0.004295
## F-statistic: 2.997 on 1 and 462 DF,  p-value: 0.08407
```

```
#With a p-value of 0.08407, the ham.trend model is not significant itself.

#Plot the trend line for ham.ts (run both lines together)
plot(ham.ts)
abline(ham.trend,col='red')
```

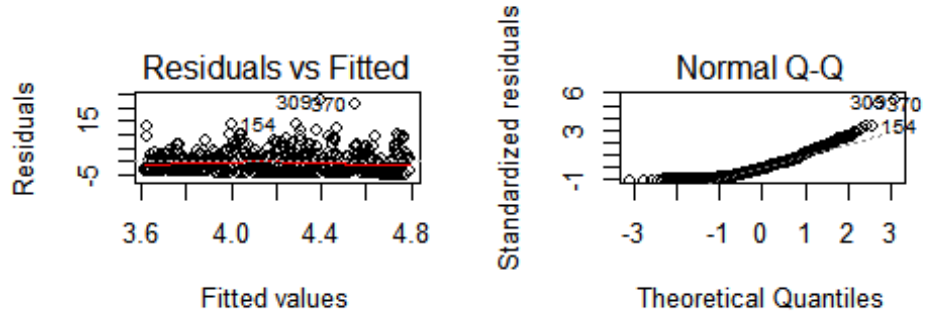


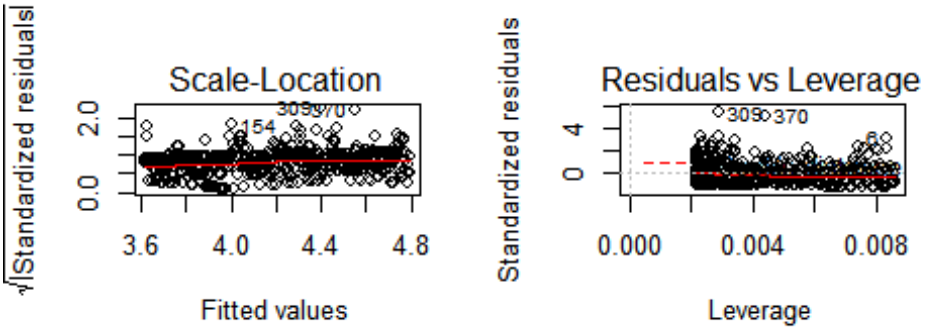

```
#The trend line increases slightly throughout the data, but no specific trend
is found.

# ACF & PACF OF HAM TREND

#We analyze the residuals of the ham.trend through ACF and PACF

#Store the residuals
e.ts.hamtrend<-ts(ham.trend$residuals)
```

```
par(mfrow=c(1,2))
acf(e.ts.hamtrend, main="Diff ACF of Residuals from ham.trend")
pacf(e.ts.hamtrend,main="Diff PACF of Residuals from ham.trend")
```



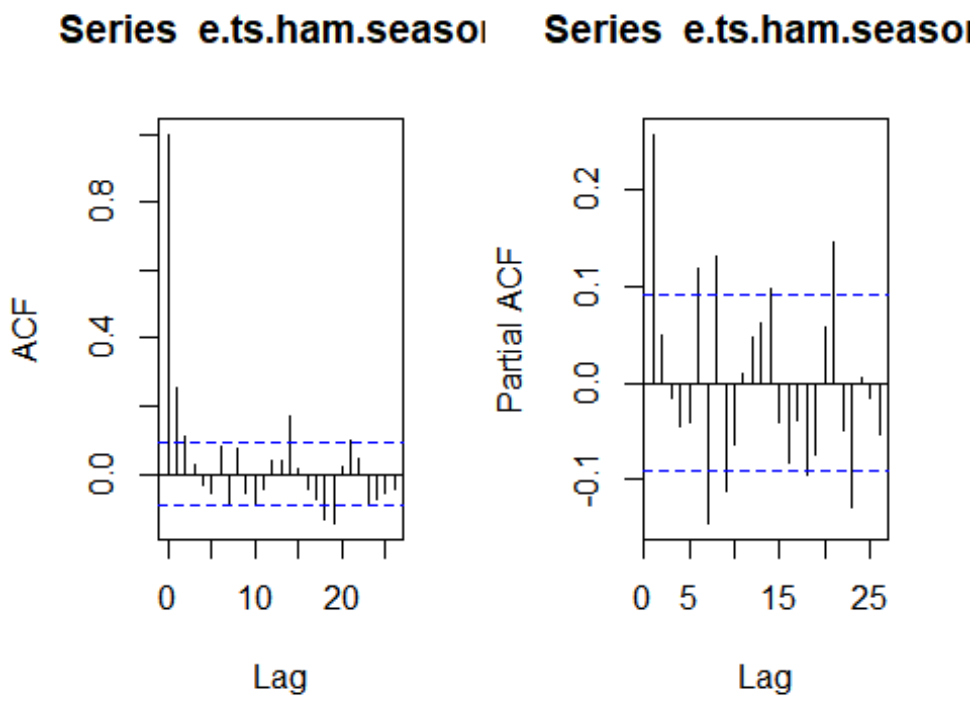

```
par(mfrow=c(1,1))
# ACF residuals for ham.trend shows sinosoidal seasonality of about 7 days,
and there are significant cutoffs all over lags 1-25.
# PACF residuals for ham.trend has significant cutoffs at lag 1, 2, 6, 7, 9,
14,and 15.
# The ham trend is stationary because there is no evidence of linear decay.

# Diagnostic plots for ham.trend
par(mfrow=c(2,2))
plot(ham.trend, labels.id = NULL)
```



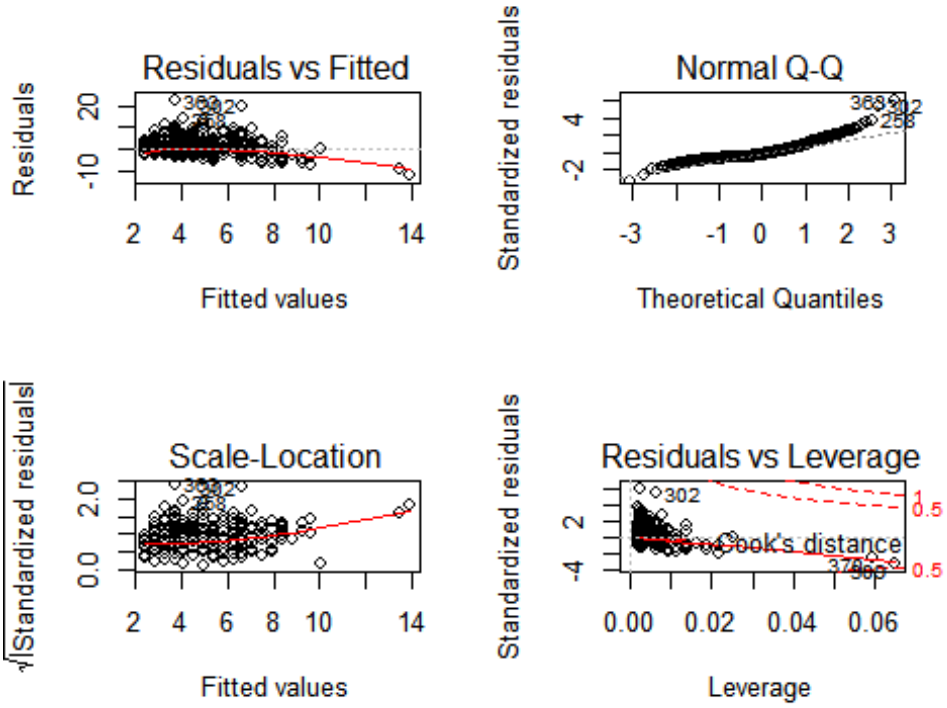



```
par(mfrow=c(1,1))
# The diagnostic plots show lack of fit with influential points 309 and 370.

#MODEL OF SEASONALITY

#Since seasonality was clearly found in the ham data earlier through the
periodograms, models of seasonality can be created for the ham data.

#First, we create the intervals for the ham data using 7 days, as this is the
period found from the periodograms.
ham.day <- time.ham %% 7
ham.day <-as.factor(time.ham %% 7)

# Build ham.season which predicts ham.ts with a 7 day lag due to the period
found earlier in the data set
ham.ts.7 = ts.intersect(ham.ts, ham7=lag(ham.ts,-7), dframe=TRUE)
ham.season <- lm(ham.ts~ ham7, data=ham.ts.7)

# Use summary to find if ham.season is significant
summary(ham.season)

##
## Call:
## lm(formula = ham.ts ~ ham7, data = ham.ts.7)
##
## Residuals:
```

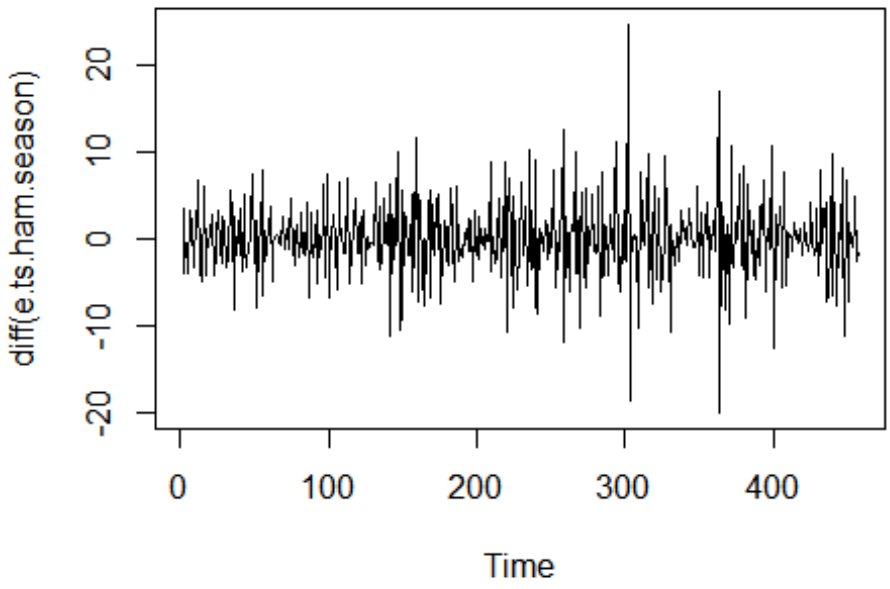
```
##      Min       1Q   Median       3Q      Max
## -11.8973  -2.4053  -0.9591   1.7434  22.3178
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.40533    0.24932   9.647   <2e-16 ***
## ham7         0.42563    0.04162  10.227   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.788 on 455 degrees of freedom
## Multiple R-squared:  0.1869, Adjusted R-squared:  0.1851
## F-statistic: 104.6 on 1 and 455 DF,  p-value: < 2.2e-16

# Ham.season is significant with low p-value of <2.2e-16.

#ACF AND PACF OF HAM SEASON

#Store residuals
e.ts.ham.season<-ts(ham.season$residuals)

#ACF and PACF for ham.season
par(mfrow=c(1,2))
acf(e.ts.ham.season)
pacf(e.ts.ham.season)
```



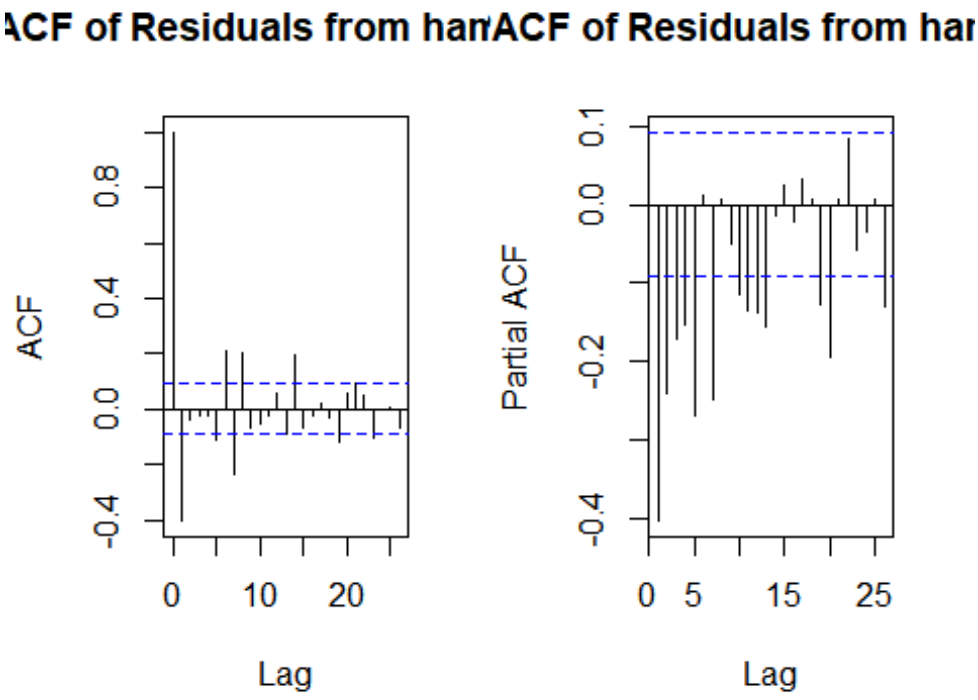
```r
par(mfrow=c(1,1))
# ACF shows sinosoidal decay, and significant cutoffs at lag 1, 2, 14, 18,
19, and 21.
# PACF shows significant cutoffs at lag 1, 6, 7, 8, 9, 14, 18, 21, and 23.
# Ham dataset is stationary as there is no linear decay in ACF in the data.

# Diagnostic plots for ham.season
par(mfrow=c(2,2))
plot(ham.season, labels.id = NULL)
```




```r
par(mfrow=c(1,1))
# The diagnostic plots show lack of fit with influential points 302 and 363.

#The ham.season still has the structure to be modeled.

# We need to see if we need to consider a first order difference of our
residuals.
#Store residuals of differences
plot(diff(e.ts.ham.season))
```




```r
#We must plot the ACF and PACF
par(mfrow=c(1,2))
acf(diff(e.ts.ham.season), main="Diff ACF of Residuals from ham.season")
pacf(diff(e.ts.ham.season),main="Diff PACF of Residuals from ham.season")
```




```
par(mfrow=c(1,1))
#From the ACF and PACF, we do not need to analyze the first order difference
of residuals for ham.season.


#MODEL WITH TREND AND SEASON

#We can model the ham data with both trend and season. We can model the
seasonality for ham data set using dummy variables, use day of the week as
the interval for the model.

#Encode the days of the week for the ham.ts
Day <- rep(NA, length(ham.ts))
Day[which((time.ham %% 7)    == 1)] <- "Th"
Day[which((time.ham %% 7)    == 2)] <- "F"
Day[which((time.ham %% 7)    == 3)] <- "Sa"
Day[which((time.ham %% 7)    == 4)] <- "S"
Day[which((time.ham %% 7)    == 5)] <- "M"
Day[which((time.ham %% 7)    == 6)] <- "T"
Day[which((time.ham %% 7)    == 0)] <- "W"
Day <- as.factor(Day)

contrasts(Day)

##    M S Sa T Th W
## F  0 0  0 0  0 0
```

```
## M  1 0  0 0  0 0
## S  0 1  0 0  0 0
## Sa 0 0  1 0  0 0
## T  0 0  0 1  0 0
## Th 0 0  0 0  1 0
## W  0 0  0 0  0 1

#The base case for the days of the week is Friday.

#Build model with both trend and season
ham.trendseason<-lm(ham.ts~time.ham+Day)

#Use summary to find if model is significant
summary(ham.trendseason)

##
## Call:
## lm(formula = ham.ts ~ time.ham + Day)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7062 -2.0896 -0.4031  1.0864 21.0085
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.752069   0.515480   9.219  < 2e-16 ***
## time.ham     0.002409   0.001220   1.975   0.0489 *
## DayM        -0.206168   0.610439  -0.338   0.7357
## DayS        -4.718910   0.610441  -7.730 6.91e-14 ***
## DaySa       -4.974076   0.610446  -8.148 3.58e-15 ***
## DayT         0.927787   0.610439   1.520   0.1292
## DayTh        0.494947   0.608141   0.814   0.4161
## DayW         0.682953   0.610441   1.119   0.2638
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.52 on 456 degrees of freedom
## Multiple R-squared:  0.3229, Adjusted R-squared:  0.3125
## F-statistic: 31.06 on 7 and 456 DF,  p-value: < 2.2e-16

#Model is clearly significant with low p-value of <2.2e-16.
```

b) Describe how the results from your graphical analysis inform model choice.

From the graphical analysis, we can clearly see that there is seasonality present within the ham time series data with a period of about 7 days. We can see that ham.trend is not significant, while both ham.season and ham.trendseason are significant models. Therefore, we choose to use ham.season for modeling purposes because it best addresses the ham time series dataset and does not incorporate trend as it is not significant.

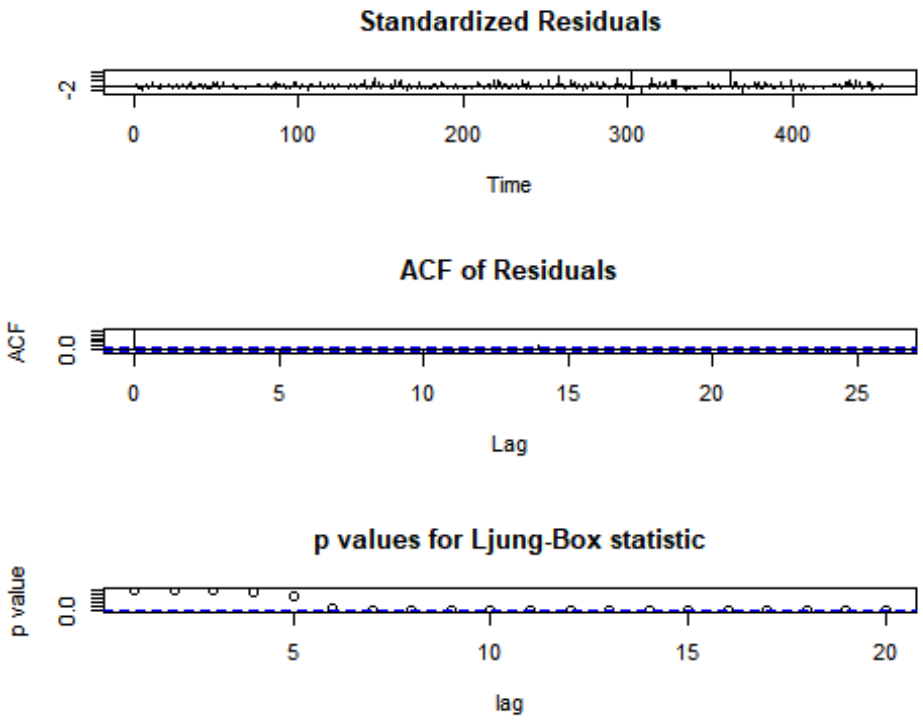
Therefore, in modeling ham.season, we decided to create a few hand-picked models from the ACF and PACF graphs, as well as an autogenerated model to compare. From the ACF and PACF graph from ham.season, we can see that the ACF represents sinusoidal decay and begins to tail off after 1 lags, where it becomes less significant. Also, the PACF shows that it cuts-off after 2 lags as well, where it is no longer significant. Since there is no linear decay in the ACF, it is a stationary process. Therefore, we concluded that the best values for p = 2, corresponding to the PACF, and the best value for q = 1, corresponding to the ACF.

Thus, it is best to begin building hand-picked AR and ARMA models for comparison. The AR models of form AR(p) include AR(2) for autoaggressive modeling. Next, an the ARMA model was built to reflect p=2 and q=1. ARIMA models were also manually built to reflect p=2, q=1 and d = 1, as well as d=2. Finally, an autogenerated auto.arima model was also built that would hopefully reflect the p=2 and q=1 parameters.

c) Based on these findings, build and evaluate models to predict the number of ham e-mails on a given day. Describe how you assess your models, diagnose problems with your models, and how you make adjustments based on these assessments. You should build one model where you select the model order yourself and one using automated selection techniques.

```
#5 models were picked: 4 manually built and one auto-generated. AIC and BIC
were used to compare models and Ljung-Box statistic used to evaluate
diagnostics of if models were adequate or not. Also, forecasting of the final
week of ham data was also taken into account as well to see if predicted
values would match the actual values.

#AR MODEL

# Estimated AR model with lag of 7, and p = 2
ham.ar2 <- arima(e.ts.ham.season, order=c(2,0,0))
summary(ham.ar2)

##
## Call:
## arima(x = e.ts.ham.season, order = c(2, 0, 0))
##
## Coefficients:
##          ar1     ar2  intercept
##       0.2437  0.0486    -0.0017
## s.e.  0.0467  0.0467     0.2409
##
## sigma^2 estimated as 13.31:  log likelihood = -1240,  aic = 2488.01
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.0005163924 3.648582 2.695575 106.8791 142.7257 0.8201379
##                      ACF1
## Training set 0.001223872

AIC(ham.ar2)
```

```
## [1] 2488.008

#AIC = 2488.008
BIC(ham.ar2)

## [1] 2504.507

#BIC = 2504.507
tsdiag(ham.ar2,gof.lag=20)
```




```
#Diagnostics: After lag of 5, p-value is significant and null hypothesis is
rejected. Model is not adequate after lag of 5.


#ARMA MODEL
#Estimated ARMA model with lag of 7, and p = 2, q = 1
ham.arma21 <- arima(e.ts.ham.season, order=c(2,0,1))
summary(ham.arma21)

##
## Call:
## arima(x = e.ts.ham.season, order = c(2, 0, 1))
##
## Coefficients:
##           ar1     ar2     ma1  intercept
##       -0.6631  0.2905  0.9235    -0.0043
## s.e.   0.0598  0.0449  0.0445     0.2367
```

```
##
## sigma^2 estimated as 13.06:  log likelihood = -1235.8,  aic = 2481.59
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.001616457 3.614533 2.666262 100.9869 146.1343 0.8112192
##                      ACF1
## Training set 0.003864752
```

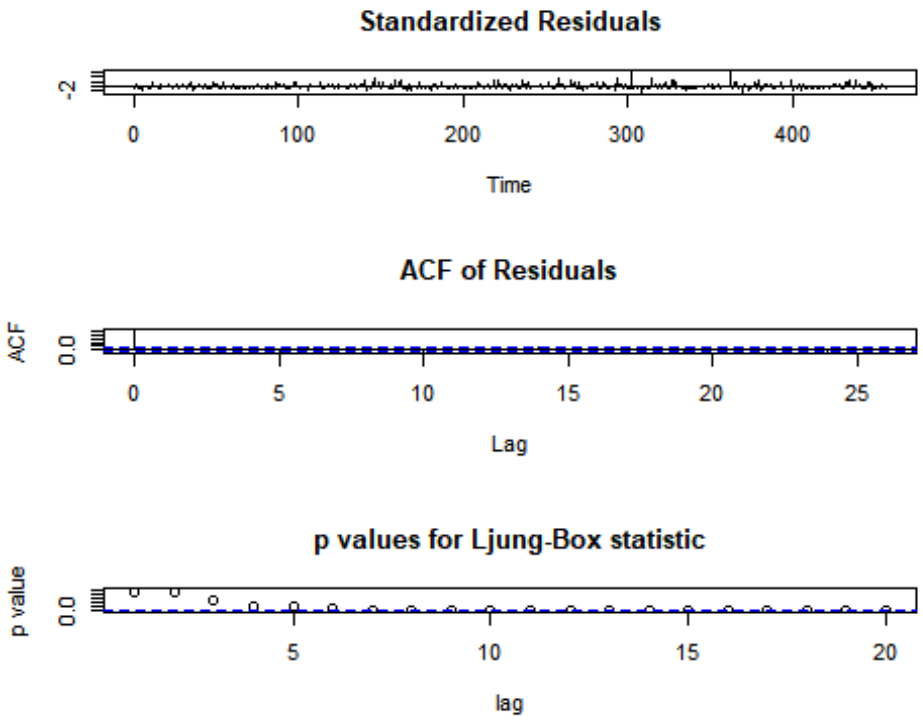
```
AIC(ham.arma21)
```

```
## [1] 2481.592
```

```
#AIC = 2481.592
BIC(ham.arma21)
```

```
## [1] 2502.215
```

```
#BIC = 2502.215
tsdiag(ham.arma21,gof.lag=20)
```




```
#Diagnostics: After lag of 5, p-value is significant and null hypothesis is
rejected. Model is not adequate after lag of 5.

#ARIMA MODEL

# Estimated ARIMA model with lag of 7, and p = 2, d=1, q = 1
```

```r
ham.arima211 <- arima(e.ts.ham.season, order=c(2,1,1))
summary(ham.arima211)
```

```
##
## Call:
## arima(x = e.ts.ham.season, order = c(2, 1, 1))
##
## Coefficients:
##          ar1     ar2      ma1
##       0.2460  0.0509  -1.0000
## s.e.  0.0468  0.0468   0.0116
##
## sigma^2 estimated as 13.34:  log likelihood = -1240.5,  aic = 2489.01
##
## Training set error measures:
##                      ME   RMSE      MAE      MPE     MAPE      MASE
## Training set 0.2285714 3.6486 2.650935 115.3351 147.3189 0.8065558
##                       ACF1
## Training set -0.004324288
```

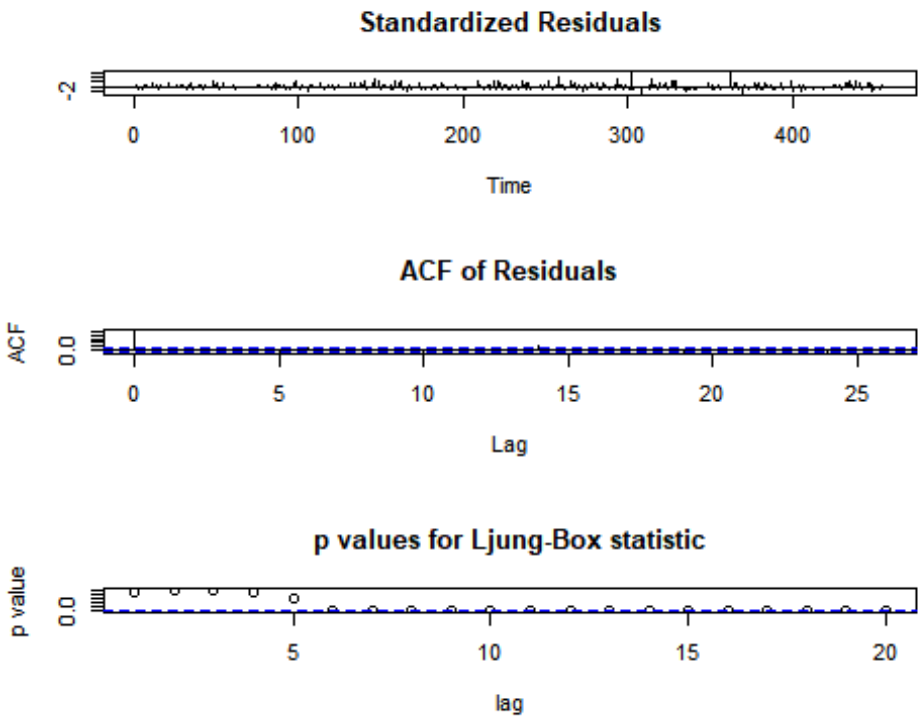
```r
AIC(ham.arima211)
```

```
## [1] 2489.009
```

```r
#AIC = 2489.009
BIC(ham.arima211)
```

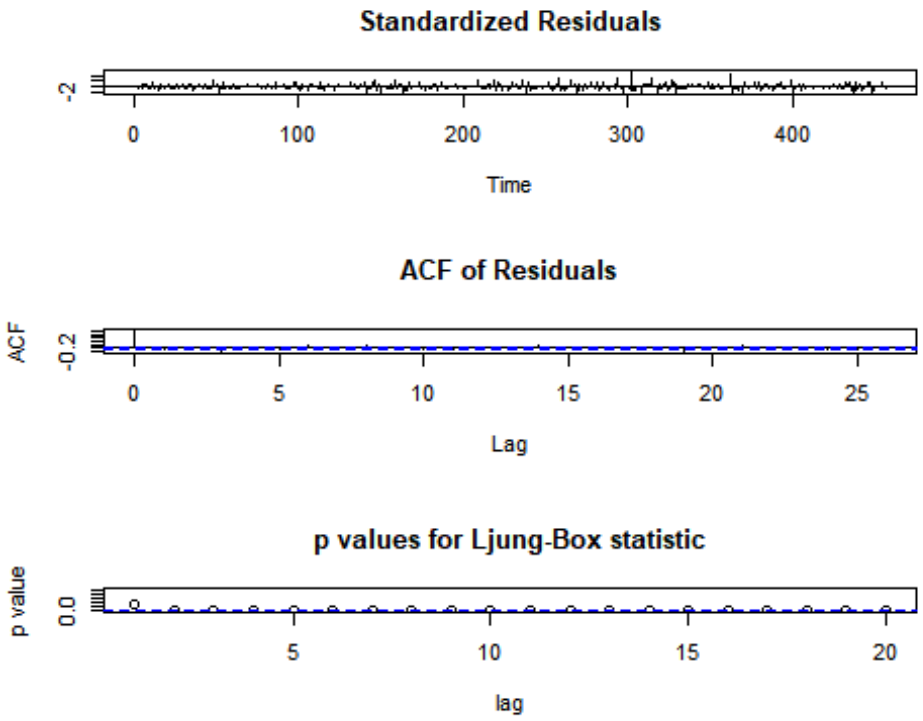
```
## [1] 2505.499
```

```r
#BIC = 2505.499
tsdiag(ham.arima211,gof.lag=20)
```




```
#Diagnostics for Ljung-Box Statistic: after lag of 5, p-value is significant
and null hypothesis is rejected. Model is not adequate after lag of 5.

#2ND ARIMA MODEL

# Estimated ARIMA model with lag of 7, and p = 2, d=2, q = 1
ham.arima221 <- arima(e.ts.ham.season, order=c(2,2,1))
summary(ham.arima221)

##
## Call:
## arima(x = e.ts.ham.season, order = c(2, 2, 1))
##
## Coefficients:
##           ar1      ar2      ma1
##       -0.4963  -0.2380  -1.0000
## s.e.   0.0455   0.0455   0.0055
##
## sigma^2 estimated as 16.84:  log likelihood = -1291.8,  aic = 2591.6
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.003187284 4.094879 2.969248 124.1295 243.7118 0.9034039
##                      ACF1
## Training set -0.04168803

AIC(ham.arima221)
```

```
## [1] 2591.603

#AIC=2591.603
BIC(ham.arima221)

## [1] 2608.084

#BIC=2608.084
tsdiag(ham.arima221,gof.lag=20)
```




```
#Diagnostics for Ljung-Box Statistic: after lag of 1, p-value is significant and null hypothesis is rejected. Model is not very adequate and less adequate than other models.

#AUTO ARIMA MODEL
ham.season.auto <- auto.arima(e.ts.ham.season, trace = TRUE)

##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : 2492.774
##  ARIMA(0,0,0) with non-zero mean : 2516.201
##  ARIMA(1,0,0) with non-zero mean : 2487.844
##  ARIMA(0,0,1) with non-zero mean : 2491.359
##  ARIMA(0,0,0) with zero mean     : 2514.183
##  ARIMA(2,0,0) with non-zero mean : 2489.473
##  ARIMA(1,0,1) with non-zero mean : 2489.041
##  ARIMA(2,0,1) with non-zero mean : 2482.257
```

```
##  ARIMA(2,0,1) with zero mean     : 2480.214
##  ARIMA(1,0,1) with zero mean     : 2487.006
##  ARIMA(3,0,1) with zero mean     : 2488.76
##  ARIMA(2,0,0) with zero mean     : 2487.438
##  ARIMA(2,0,2) with zero mean     : 2490.72
##  ARIMA(1,0,0) with zero mean     : 2485.818
##  ARIMA(3,0,2) with zero mean     : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(2,0,1) with zero mean     : 2479.681
##
##  Best model: ARIMA(2,0,1) with zero mean
```

```
summary(ham.season.auto) #Creates ARIMA model with (2,0,1) with zero mean
```

```
## Series: e.ts.ham.season
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
##           ar1     ar2     ma1
##       -0.6623  0.2906  0.9228
## s.e.   0.0600  0.0448  0.0447
##
## sigma^2 estimated as 13.15:  log likelihood=-1235.8
## AIC=2479.59   AICc=2479.68   BIC=2496.09
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.001435362 3.614544 2.666966 100.9051 146.0924 0.8114335
##                      ACF1
## Training set 0.003771379
```

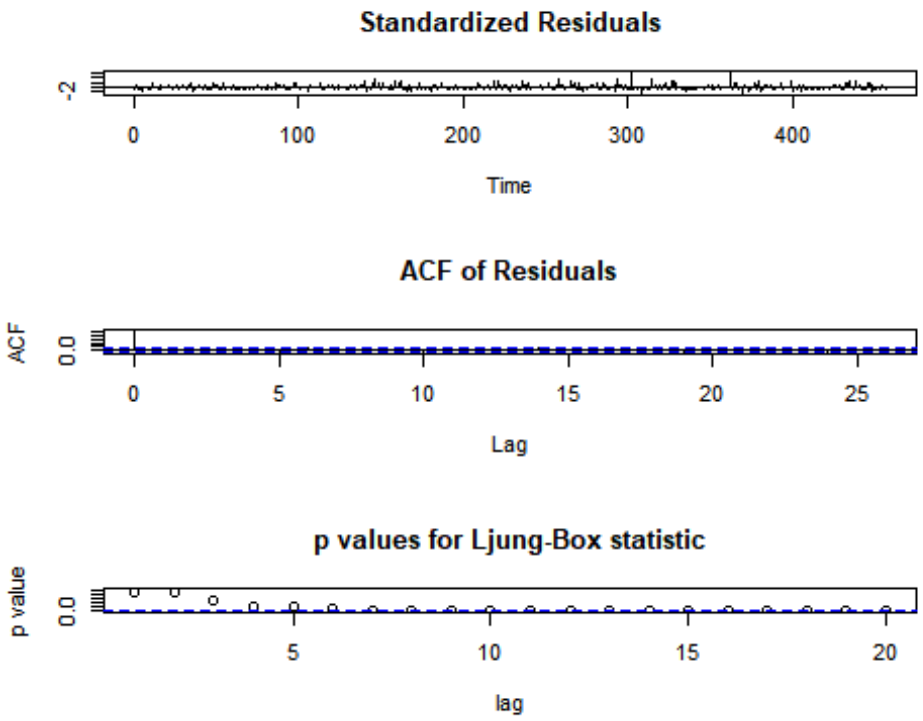
```
AIC(ham.season.auto)
```

```
## [1] 2479.592
```

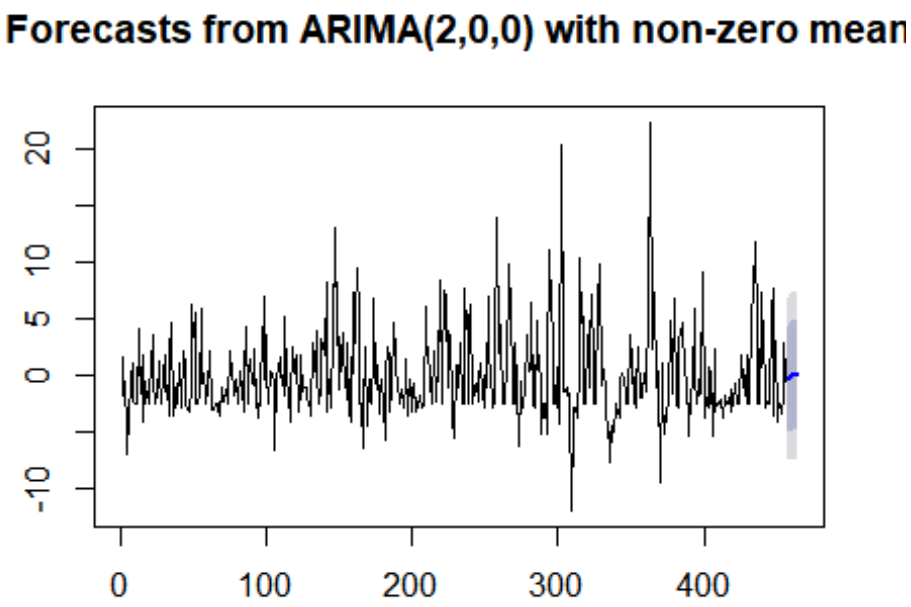
```
#AIC = 2479.592
BIC(ham.season.auto)
```

```
## [1] 2496.091
```

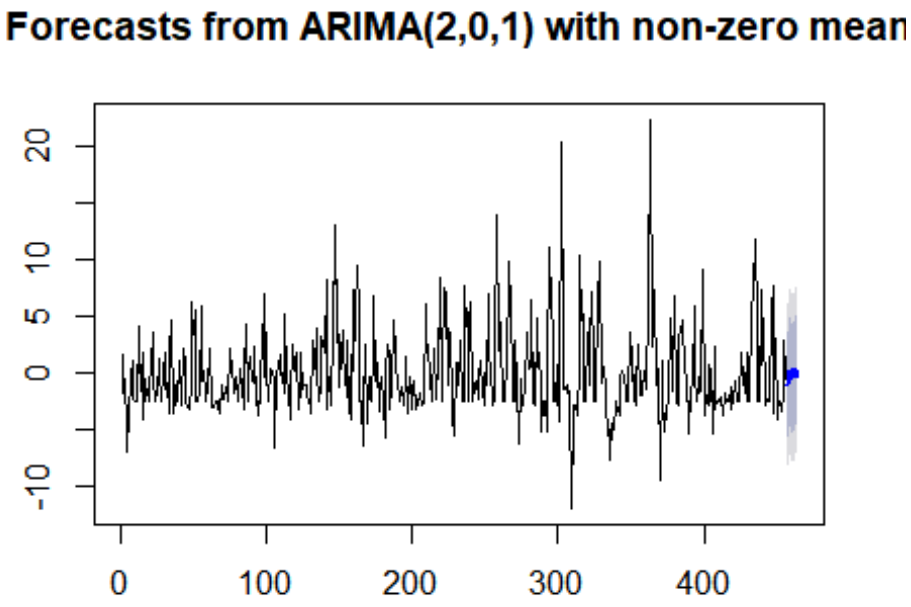
```
#BIC = 2496.091
tsdiag(ham.season.auto,gof.lag=20)
```



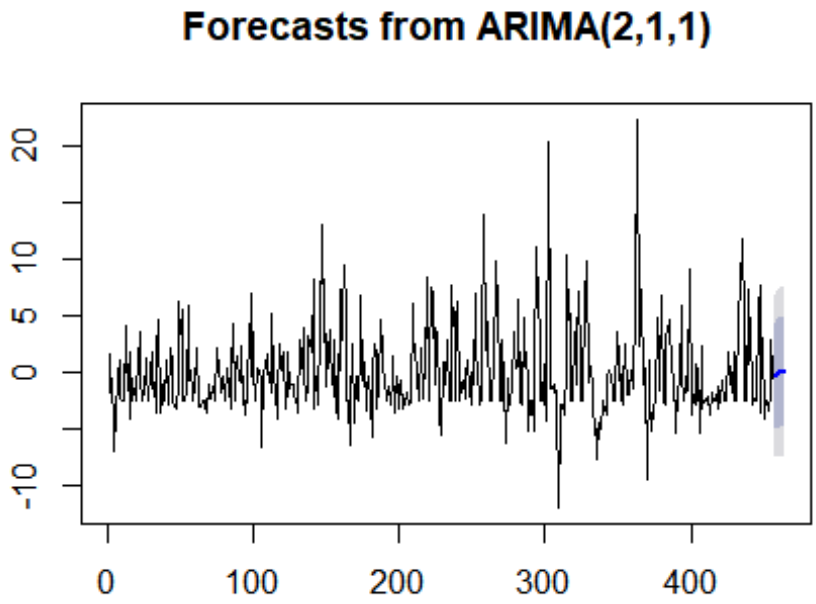

```
#Diagnostics for Ljung-Box Statistic: after lag of 5, p-value is significant
and null hypothesis is rejected. Model is not adequate after lag of 5.

#Based on AIC, BIC, and Diagnostics, ham.season.auto performs the best
because of low AIC, BIC, and best diagnostics.

#FORECASTING

# Prediction performance

#Forecast the next 7 days of ham with the forecast option for each model
ham.ar2.forecast<-forecast(ham.ar2,h=7)
ham.arma21.forecast<-forecast(ham.arma21,h=7)
ham.arima211.forecast<-forecast(ham.arima211,h=7)
ham.arima221.forecast<-forecast(ham.arima221,h=7)
ham.season.auto.forecast<-forecast(ham.season.auto,h=7)

#View plots of the forecasts of each model
plot(ham.ar2.forecast)
```



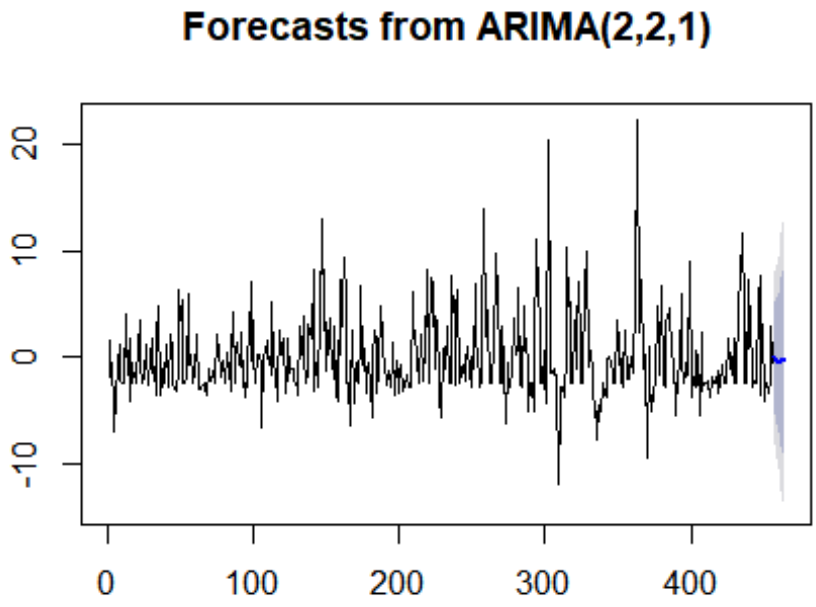

```
plot(ham.arma21.forecast)
```



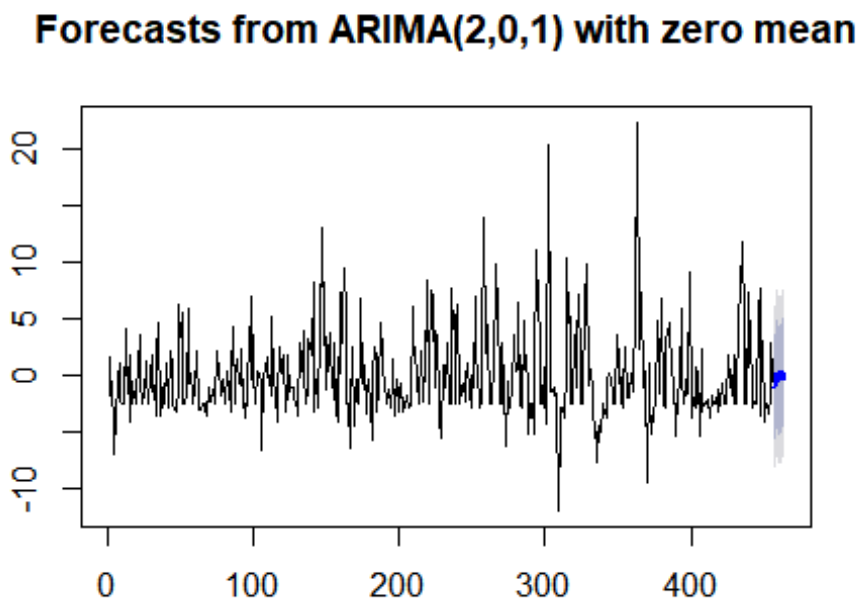

```
plot(ham.arima211.forecast)
```




```
plot(ham.arima221.forecast)
```




```
plot(ham.season.auto.forecast)
```




```r
#Seems that auto model may have best forecast

# Create test set from ham data set with last 7 days
# The next week or the test period in days
next.week.time<-c((length(ham.ts)-6):(length(ham.ts)))
# Create the test data frame
next.week<-data.frame(time.ham = next.week.time, count =
ham.ts[next.week.time])
# Create the actual time series for the test period
next.week.ts <- ham.ts[next.week.time]
next.week.ts<-ts(next.week$count)

#Create each of the predictions for the next week for each of the 5 models
next.week.prediction.ar2<-
predict(ham.trend,newdata=next.week)+forecast(ham.ar2,h=7)$mean
next.week.prediction.arma21<-
predict(ham.trend,newdata=next.week)+forecast(ham.arma21,h=7)$mean
next.week.prediction.arima211<-
predict(ham.trend,newdata=next.week)+forecast(ham.arima211,h=7)$mean
next.week.prediction.arima221<-
predict(ham.trend,newdata=next.week)+forecast(ham.arima221,h=7)$mean
next.week.prediction.season.auto<-
predict(ham.trend,newdata=next.week)+forecast(ham.season.auto,h=7)$mean

# MSE of predictions
mean((next.week.prediction.ar2-next.week$count)^2) #9.078856
```

```
## [1] 9.078856
```

```
mean((next.week.prediction.arma21-next.week$count)^2) # 8.571088
```

```
## [1] 8.571088
```

```
mean((next.week.prediction.arima211-next.week$count)^2) #9.560446
```

```
## [1] 9.068319
```

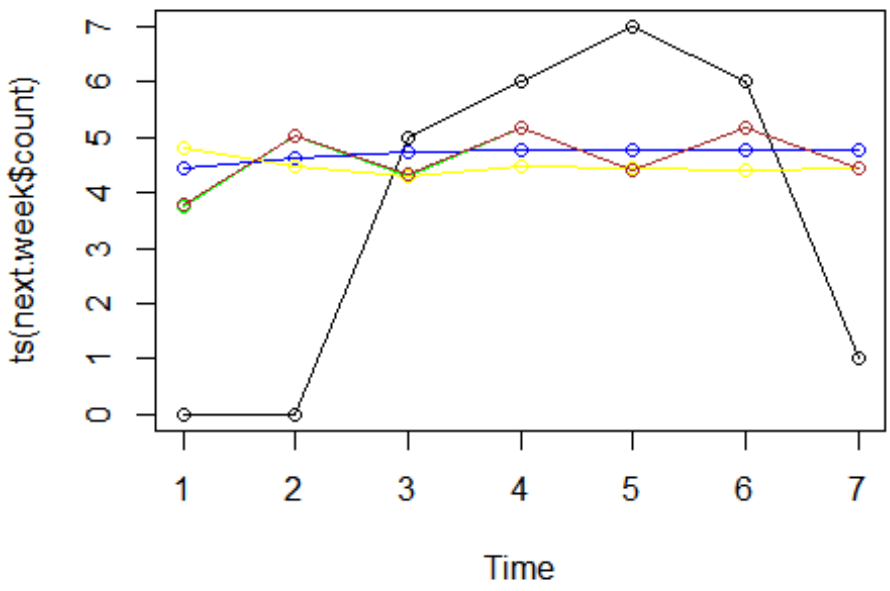
```
mean((next.week.prediction.arima221-next.week$count)^2) # 9.560446
```

```
## [1] 9.560446
```

```
mean((next.week.prediction.season.auto-next.week$count)^2) #8.578898
```

```
## [1] 8.578898
```

```
#Clear that ARMA model with p=2 and q=1 has lowest  MSE, but auto.arima model
has next lowest MSE value.

# Using plots to compare the predicted value and real values for the last
week of data
plot(ts(next.week$count),type='o')
lines(ts(next.week.prediction.ar2),col='red',type='o')
lines(ts(next.week.prediction.arma21),col='green',type='o')
lines(ts(next.week.prediction.arima211),col='blue',type='o')
lines(ts(next.week.prediction.arima221),col='yellow',type='o')
lines(ts(next.week.prediction.season.auto),col='brown',type='o')
```




```
#It is clear from the graph that the brown model, the auto.arima model,
performs closest to the actual values and that the green model, ARMA model
with p = 2 and q=1, only differs in this near the end of the forecasting.

#Problems with models: Many models become inadequate in the Ljung-Box
Statistic diagnostic after a lag of 5.
#How adjust models: Since the models are being evaluated in a rather
simplistic way, there is not much adjustment of modeling that can be done
within this report. However, further investigative analysis can be done to
increase the model's adequacy for lags past 5.
```

d) Which model do you recommend and why?

From the analysis done above, the best model is the auto generated auto.arima model, which creates a ARIMA model with p=2, d=0, and q=1 (2,0,1) with a zero mean. This model is recommended because it has the lowest AIC and BIC of all of the candidate models, with AIC = 2479.592 and BIC = 2496.091. Although, the model did have significant p-values after 5 lags in the Ljung-Box statistic, this is understandable as this is present in most of the other models and can be seen when analyzing data in a rather simplistic manner. In terms of forecasting the auto.arima generated model performed well with one of the lowest MSE values of 8.578898. The only other model with low MSE values was the hand-picked arma model (2,1) with non-zero mean and MSE values of 8.571088. In terms of visual forecasting, both of these models also performed similarly graphically in terms of predicted vs actual values for the next week. Therefore, the autogenerated arima model was chosen because of its very low AIC, BIC values, its good diagnostics, and its ability to most accurately forecast future ham emails.

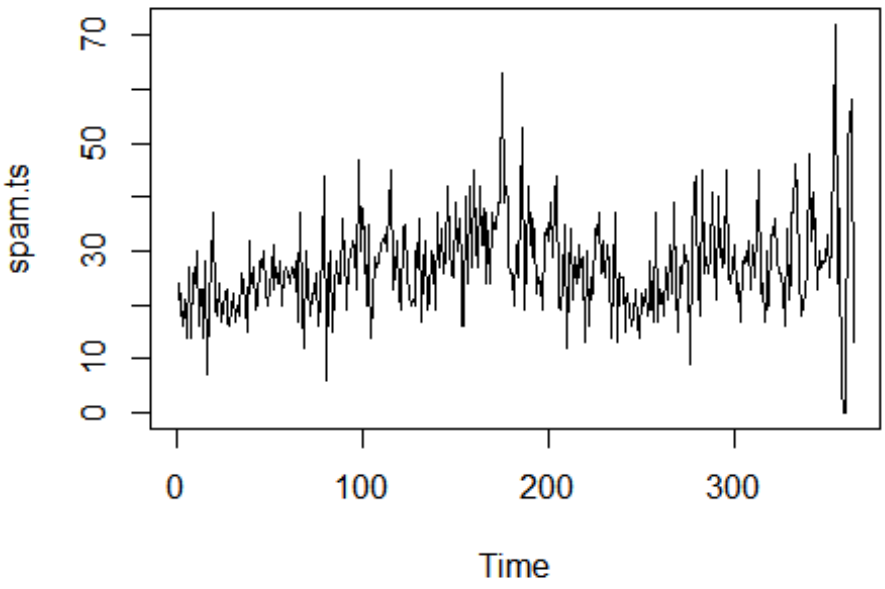
2. Spam Time Series Analysis: Build a time series filter for spam e-mail.

a)Perform a graphical analysis of spam.ts- use time series plots, periodograms, autocorrelation, and partial autocorrelation plots.

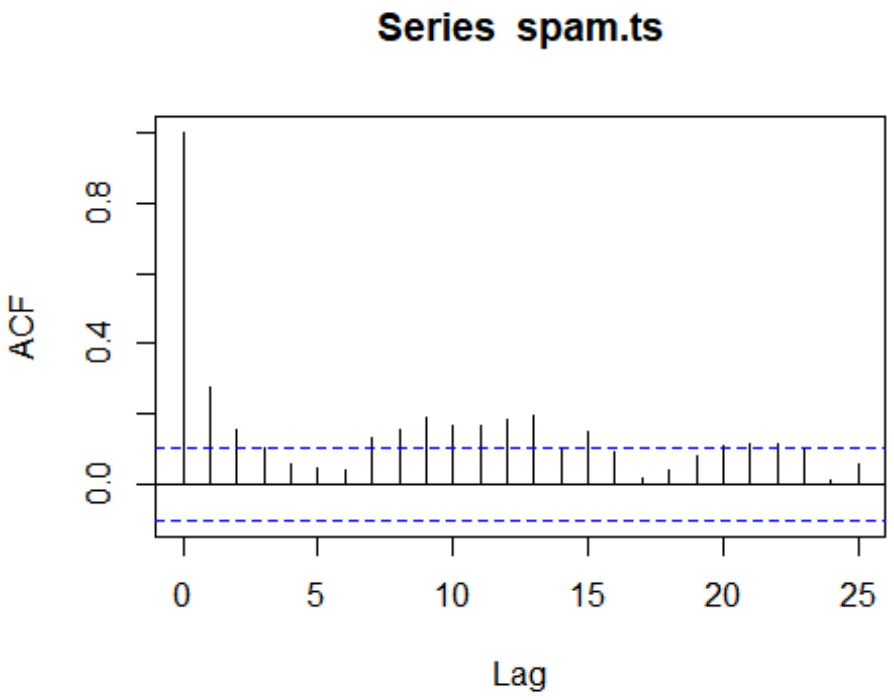
```
#Import Spam Dataset, change paths to run

spam<-read.table('C:/Users/student/Desktop/Train
Data/spam_ts.csv',header=T,sep=',')

#Displays time series data - sequence of data that have been observed in
successive order at different points in time

#Summarize the new spam data set

summary(spam)
```

```
##       year          month             day            count
##  Min.   :2004   Min.   : 1.000   Min.   : 1.00   Min.   : 0.00
##  1st Qu.:2004   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:22.00
##  Median :2005   Median : 7.000   Median :16.00   Median :26.00
##  Mean   :2005   Mean   : 6.525   Mean   :15.68   Mean   :27.37
##  3rd Qu.:2005   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:32.00
##  Max.   :2005   Max.   :12.000   Max.   :31.00   Max.   :72.00
```

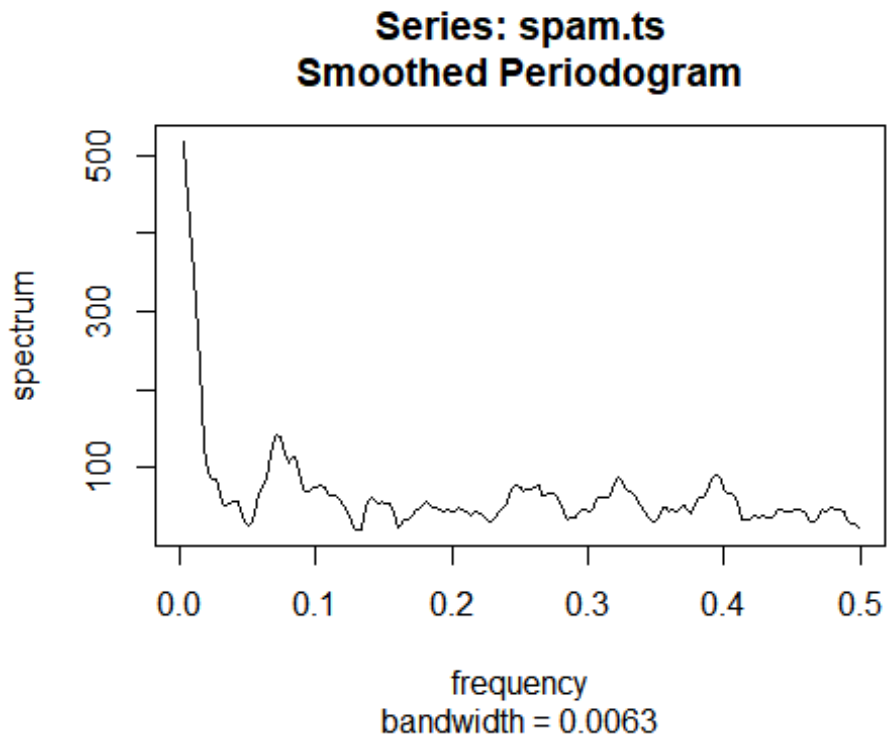
```
#Interpretation: Represents time series data; variables include year, month,
day, and count; Ranges for Year: (2004, 2005), Month: (1,12), Day: (1:31),
Count(0, 72)

#The ts() command can be used to get a time series of spam amount using spam
data from all years
spam.ts<-ts(spam$count)

#Load the forecast library, make sure to have package installed by
'install.packages("forecast")'
library(forecast)

#TIME SERIES PLOTS

#First, we must plot the spam time series created from the data
plot(spam.ts)
```




```
#Interpretation: There is nothing too obvious in the time series plot of the
spam.ts. No abnormalities.

#AUTOCORRELATION

#Use the acf() command to create the time series for spam.
acf(spam.ts)
```




```
#There seems to be sinusoidal decay and there are also significant
autocorrelations after lag=5.

#PERIODIGRAM

#This can be further explored through a periodigram. The peak in the
frequency can be analyzed in order to find the period for the spam.ts.
pg.spam<-spec.pgram(spam.ts, spans=9, demean=T, log='no') #Periodigram
```




```
max.omega.spam<-pg.spam$freq[which(pg.spam$spec==max(pg.spam$spec))] #Find
peak
1/max.omega.spam # 375 day period
```

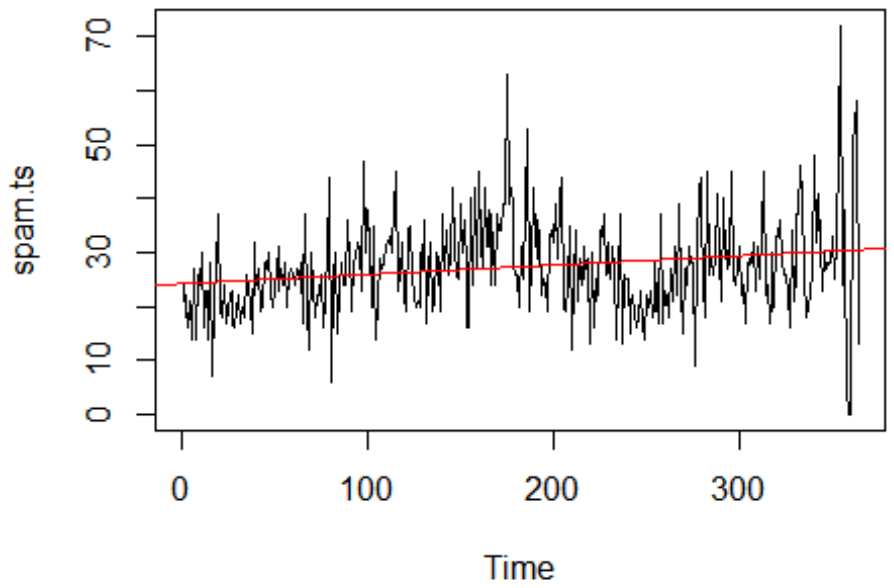
```
## [1] 375
```

```
# The seems to be no presence of seasonality in  the spam data and no
consistent period as well.

#MODEL TREND OF SPAM

#We must model the trend of spam to find if this is present in the data.

#Create a new variable time.spam which is a matrix of (length(spam.ts))
time.spam<-c(1:length(spam.ts))

#Build a new model, spam.trend which predicts spam.ts based on the time
variable, time.spam
spam.trend<-lm(spam.ts~time.spam)
summary(spam.trend)
```

```
##
## Call:
## lm(formula = spam.ts ~ time.spam)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

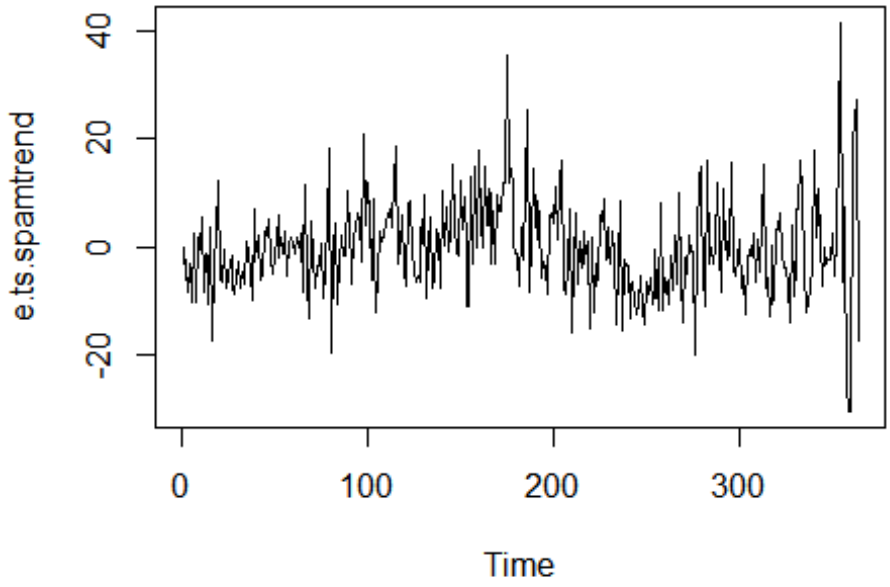
```
## -30.479  -5.633  -0.786   4.623  41.626
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.175415   0.940017  25.718  < 2e-16 ***
## time.spam    0.017509   0.004464   3.923 0.000105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 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
```

```
#With a p-value of 0.000104, the spam.trend model is significant itself, as
well as time.spam being significant with p-value of 0.000105.

#Plot the trend line for spam.ts (run both lines together)
plot(spam.ts)
abline(spam.trend,col='red')
```

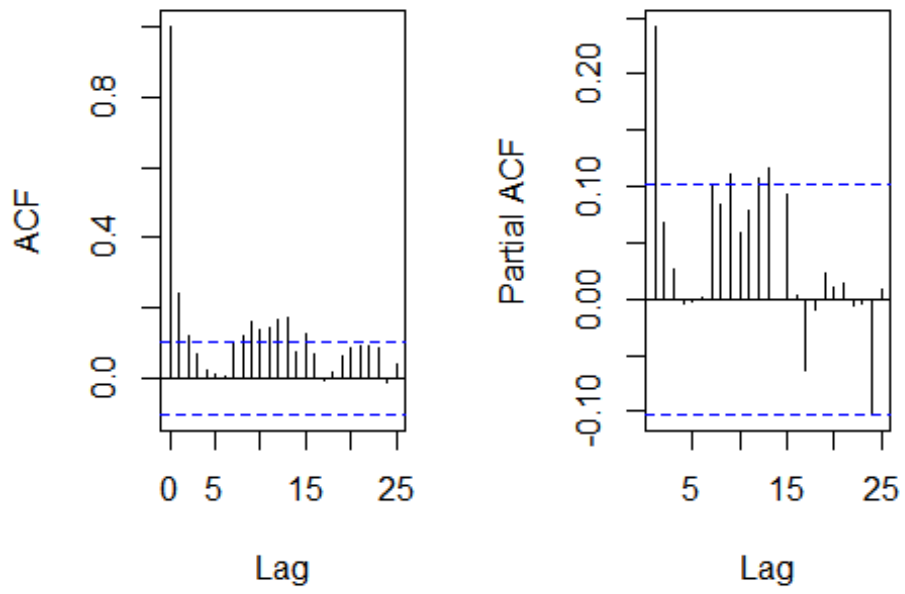



```
#The trend line increases slightly throughout the data.

# ACF & PACF OF SPAM TREND

#We analyze the residuals of the spam.trend through ACF and PACF

#Store the residuals
```

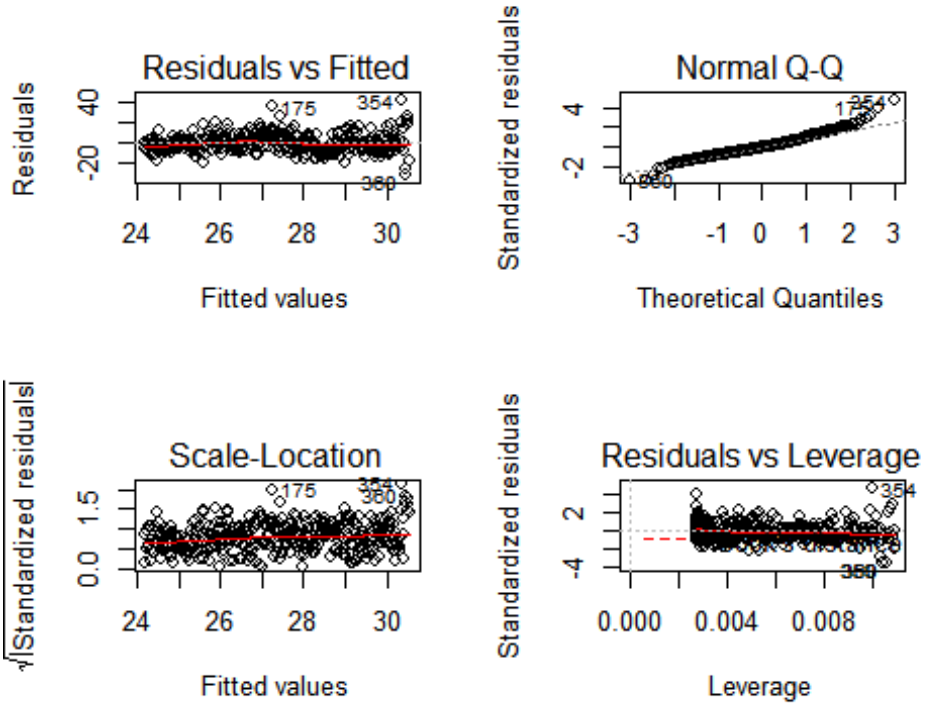
```r
e.ts.spamtrend<-ts(spam.trend$residuals)
plot(e.ts.spamtrend)
```



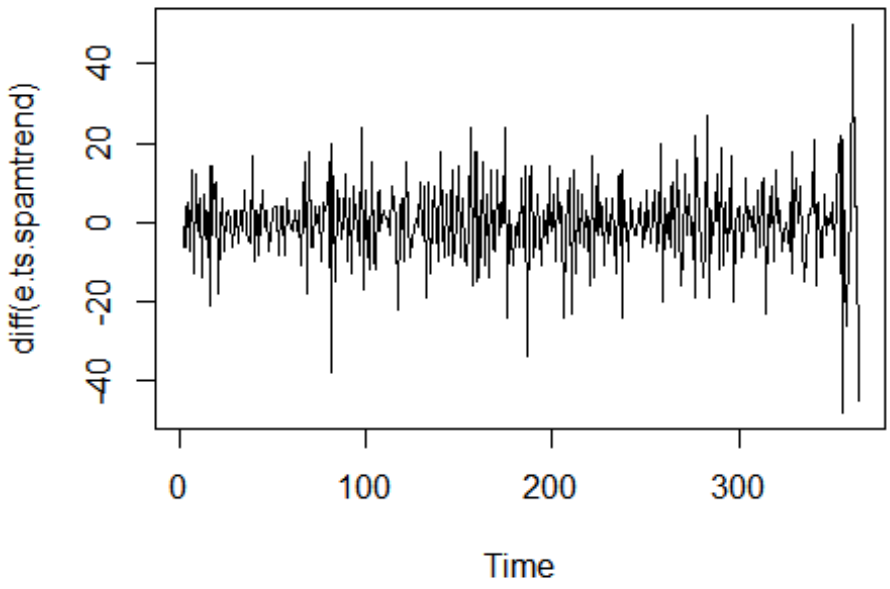

```r
#This plot of the residuals looks very similar to original data set.

#Create ACF and PACF for spam.trend
par(mfrow=c(1,2))
acf(e.ts.spamtrend, main="Diff ACF of Residuals from spam.trend")
pacf(e.ts.spamtrend,main="Diff PACF of Residuals from spam.trend")
```



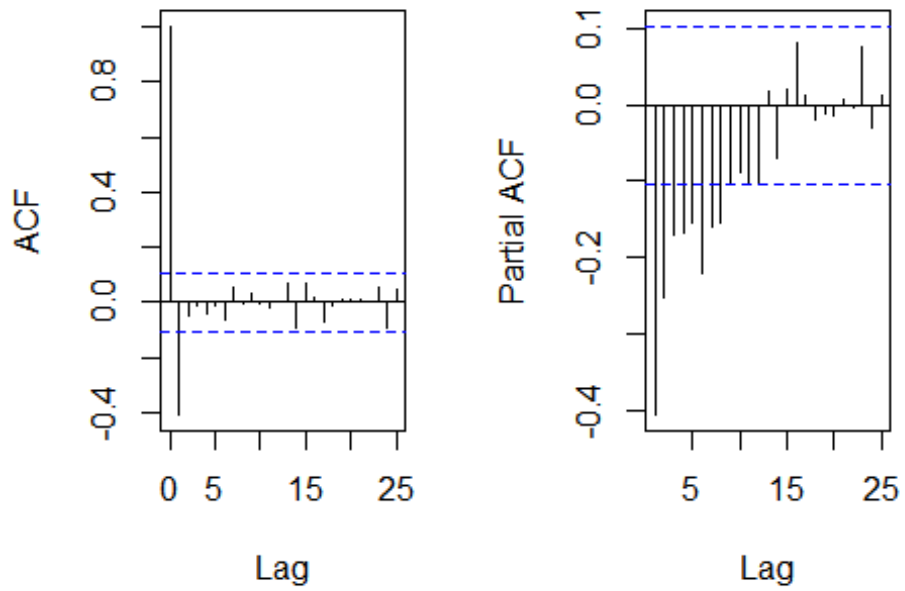

```r
par(mfrow=c(1,1))
#ACF plot for spam.trend shows sinosoidal decay, and there are significant
cutoffs at lag 1,2,8,9,10,11,12,13,15.
#PACF residuals for spam.trend has significant cutoffs at lag 1,9,12,13.

#The time series has the structure to be modeled.

# Diagnostic plots for spam.trend
par(mfrow=c(2,2))
plot(spam.trend, labels.id = NULL)
```




```r
par(mfrow=c(1,1))
# The diagnostic plots show some lack of fit, but generally seem acceptable.
There are a few points that seem as though they may be outliers, such as
observation of 175 and 354.

# Assess if first order difference of residuals must be assessed
#Plot difference of residuals
plot(diff(e.ts.spamtrend))
```




```r
#Analyze the difference between the residuals through ACF and PACF.
par(mfrow=c(1,2))
acf(diff(e.ts.spamtrend), main="Diff ACF of Residuals from spam.trend")
pacf(diff(e.ts.spamtrend),main="Diff PACF of Residuals from spam.trend")
```




```
par(mfrow=c(1,1))

#First order difference of residuals do not need to be assessed in this case.
```

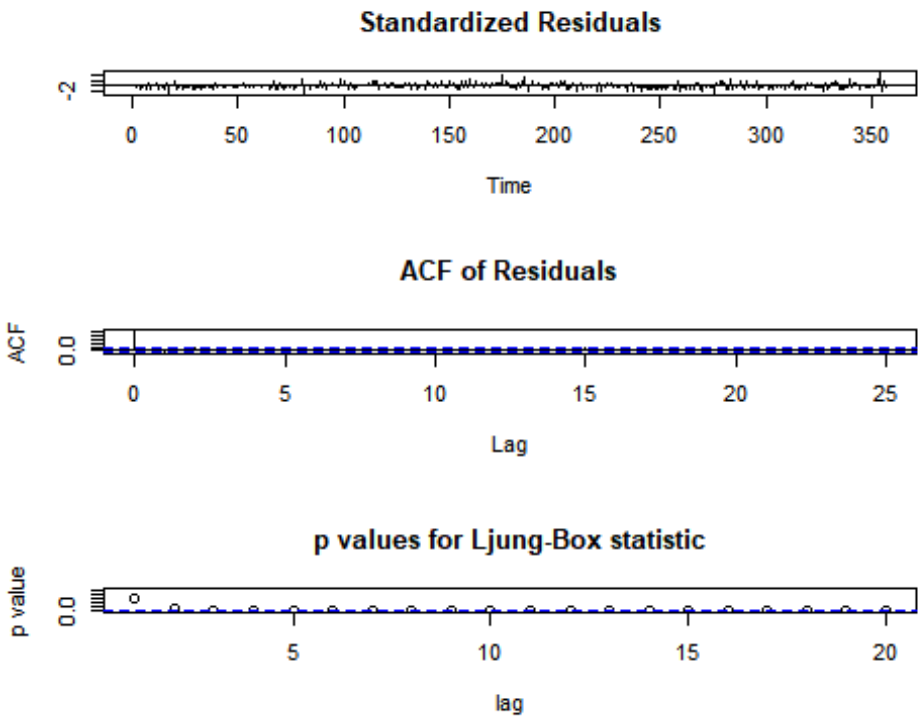
b) Describe how the results from your graphical analysis inform model choice.

From the above graphical analysis above, it appears that the trend pattern in spam is significant. Therefore, through the PACF plot of spam.trend, it can be seen that there is a roughly sinusoidal decay and that it becomes insignificant and there is a cut off after lag 1, meaning p=1. Moreover, from the ACF, it can be seen that there is somewhat of sinusoidal decay with a cut off after lag 1 as well, meaning q=1. Since the best models often arise from the least complexity, keeping these values at a minimum would be most advantageous in terms of performance metrics.

Therefore, from this information, the models should include p=1 and/or q=1 based off the spam ACF and PACF. Thus, AR, ARMA, ARIMA, and auto generated models were used to fit these parameters. Moreover, other models were built that would use all of the spam data set as well to see how these models would fair in comparison. Auto and ARMA models were generated for this whole data set.

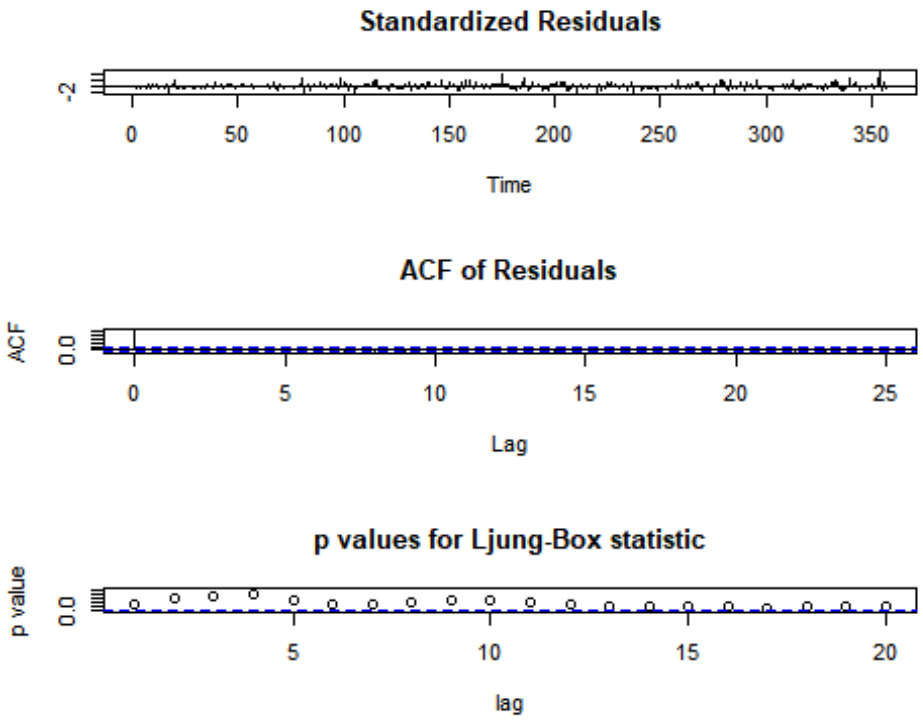
c) Based on these findings, build and evaluate models to predict the number of spam e-mails on a given day. Describe how you assess your models, diagnose problems with your models, and how you make adjustments based on these assessments. You should build one model where you select the model order yourself and one using automated selection techniques.

```r
#6 Models were built. AIC and BIC were used to compare models and Ljung-Box
statistic used to evaluate diagnostics of if models were adequate or not.
Also, forecasting of the final week of spam data was also taken into account
as well to see if predicted values would match the actual values.

##Model spam trend without last week
time.spam<-c(1:(length(spam.ts)-7))

# Get a linear model of spam vs. the index of the day

spam.trend<-lm(spam.ts[time.spam]~time.spam)

#get the residuals from the spam.trend model above and store in e.ts:

e.ts.spamtrend<-ts(spam.trend$residuals)

#AR MODEL

# Estimated AR model with p=1
spam.ar1 <- arima(e.ts.spamtrend, order=c(1,0,0))
summary(spam.ar1)

##
## Call:
## arima(x = e.ts.spamtrend, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.2157     0.0001
## s.e.  0.0516     0.5451
##
## sigma^2 estimated as 65.36:  log likelihood = -1252.69,  aic = 2511.38
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE    MAPE      MASE
## Training set 4.518368e-05 8.084403 6.229348 101.3216 199.791 0.7794791
##                     ACF1
## Training set -0.02896727

AIC(spam.ar1)

## [1] 2511.385

#AIC = 2511.385
BIC(spam.ar1)

## [1] 2523.018

#BIC = 2523.018
tsdiag(spam.ar1,gof.lag=20)
```




```
#Diagnostics: Model is mostly inadequate because low p-values after lag of 1.


#ARMA MODEL
#Estimated ARMA model with p=1, q=1
spam.arma11 <- arima(e.ts.spamtrend, order=c(1,0,1))
summary(spam.arma11)

##
## Call:
## arima(x = e.ts.spamtrend, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.9615  -0.8690     0.1069
## s.e.  0.0260   0.0468     1.3496
##
## sigma^2 estimated as 61.84:  log likelihood = -1242.99,  aic = 2493.97
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.07071521 7.863922 6.016065 73.41651 227.1488 0.7527911
##                      ACF1
## Training set 0.04713333

AIC(spam.arma11)
```

```
## [1] 2493.974

#AIC = 2493.974
BIC(spam.arma11)

## [1] 2509.485

#BIC = 2509.485
tsdiag(spam.arma11,gof.lag=20)
```




```
#Diagnostics: Model is very adequate because of high p-values for Ljung Box statistic.

#ARIMA MODEL

# Estimated ARIMA model with p = 1, d=1, q = 1
spam.arima111 <- arima(e.ts.spamtrend, order=c(1,1,1))
summary(spam.arima111)

##
## Call:
## arima(x = e.ts.spamtrend, order = c(1, 1, 1))
##
## Coefficients:
##          ar1      ma1
##       0.0708  -0.9185
## s.e.  0.0590   0.0249
```

```
##
## sigma^2 estimated as 62.46:  log likelihood = -1241.96,  aic = 2489.92
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE    MAPE      MASE
## Training set 0.2213119 7.892252 6.004068 70.16787 269.405 0.7512898
##                       ACF1
## Training set -0.004577296
```

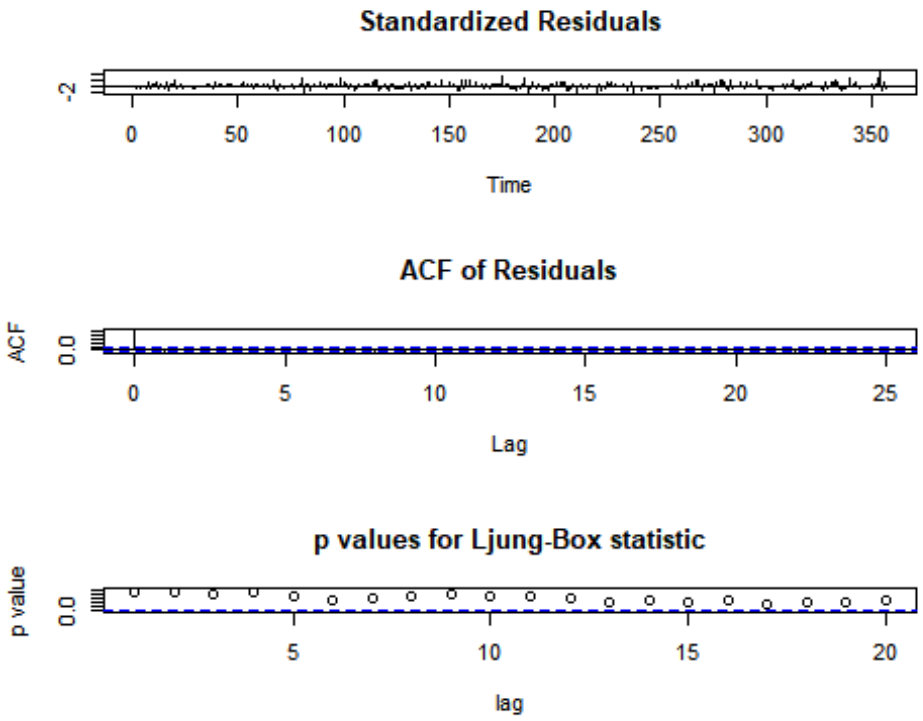
```
AIC(spam.arima111)
```

```
## [1] 2489.917
```

```
#AIC = 2489.917
BIC(spam.arima111)
```

```
## [1] 2501.541
```

```
#BIC = 2501.541
tsdiag(spam.arima111,gof.lag=20)
```




```
#Diagnostics for Ljung-Box Statistic: Model is very adequate because of high
p-values for Ljung Box statistic, accept null hypothesis.

#AUTO ARIMA MODEL
spam.auto <- auto.arima(e.ts.spamtrend, trace = TRUE, stepwise=FALSE)
```

```
##
##  ARIMA(0,0,0) with zero mean     : 2524.452
##  ARIMA(0,0,0) with non-zero mean : 2526.475
##  ARIMA(0,0,1) with zero mean     : 2512.942
##  ARIMA(0,0,1) with non-zero mean : 2514.976
##  ARIMA(0,0,2) with zero mean     : 2510.076
##  ARIMA(0,0,2) with non-zero mean : 2512.122
##  ARIMA(0,0,3) with zero mean     : 2503.404
##  ARIMA(0,0,3) with non-zero mean : 2505.461
##  ARIMA(0,0,4) with zero mean     : 2502.624
##  ARIMA(0,0,4) with non-zero mean : 2504.693
##  ARIMA(0,0,5) with zero mean     : 2504.14
##  ARIMA(0,0,5) with non-zero mean : 2506.221
##  ARIMA(1,0,0) with zero mean     : 2510.372
##  ARIMA(1,0,0) with non-zero mean : 2512.406
##  ARIMA(1,0,1) with zero mean     : 2493.034
##  ARIMA(1,0,1) with non-zero mean : 2494.973
##  ARIMA(1,0,2) with zero mean     : 2494.071
##  ARIMA(1,0,2) with non-zero mean : 2496.017
##  ARIMA(1,0,3) with zero mean     : 2495.984
##  ARIMA(1,0,3) with non-zero mean : 2497.941
##  ARIMA(1,0,4) with zero mean     : 2496.457
##  ARIMA(1,0,4) with non-zero mean : 2498.436
##  ARIMA(2,0,0) with zero mean     : 2506.009
##  ARIMA(2,0,0) with non-zero mean : 2508.052
##  ARIMA(2,0,1) with zero mean     : 2493.632
##  ARIMA(2,0,1) with non-zero mean : 2495.411
##  ARIMA(2,0,2) with zero mean     : 2494.642
##  ARIMA(2,0,2) with non-zero mean : 2496.526
##  ARIMA(2,0,3) with zero mean     : 2496.167
##  ARIMA(2,0,3) with non-zero mean : 2498.053
##  ARIMA(3,0,0) with zero mean     : 2502.284
##  ARIMA(3,0,0) with non-zero mean : 2504.337
##  ARIMA(3,0,1) with zero mean     : 2496.713
##  ARIMA(3,0,1) with non-zero mean : 2498.613
##  ARIMA(3,0,2) with zero mean     : 2497.207
##  ARIMA(3,0,2) with non-zero mean : 2499.069
##  ARIMA(4,0,0) with zero mean     : 2503.854
##  ARIMA(4,0,0) with non-zero mean : 2505.907
##  ARIMA(4,0,1) with zero mean     : 2499.508
##  ARIMA(4,0,1) with non-zero mean : 2501.249
##  ARIMA(5,0,0) with zero mean     : 2506.923
##  ARIMA(5,0,0) with non-zero mean : 2508.986
```

```
summary(spam.auto) #Creates arima model to be (1,0,1) with zero mean
```

```
## Series: e.ts.spamtrend
## ARIMA(1,0,1) with zero mean
##
## Coefficients:
```

```
##          ar1      ma1
##       0.9616  -0.8692
## s.e.  0.0260   0.0467
##
## sigma^2 estimated as 62.19:  log likelihood=-1242.99
## AIC=2491.98   AICc=2492.05   BIC=2503.61
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.1033146 7.863982 6.013328 70.05201 223.4664 0.7524486
##                     ACF1
## Training set 0.04712902
```

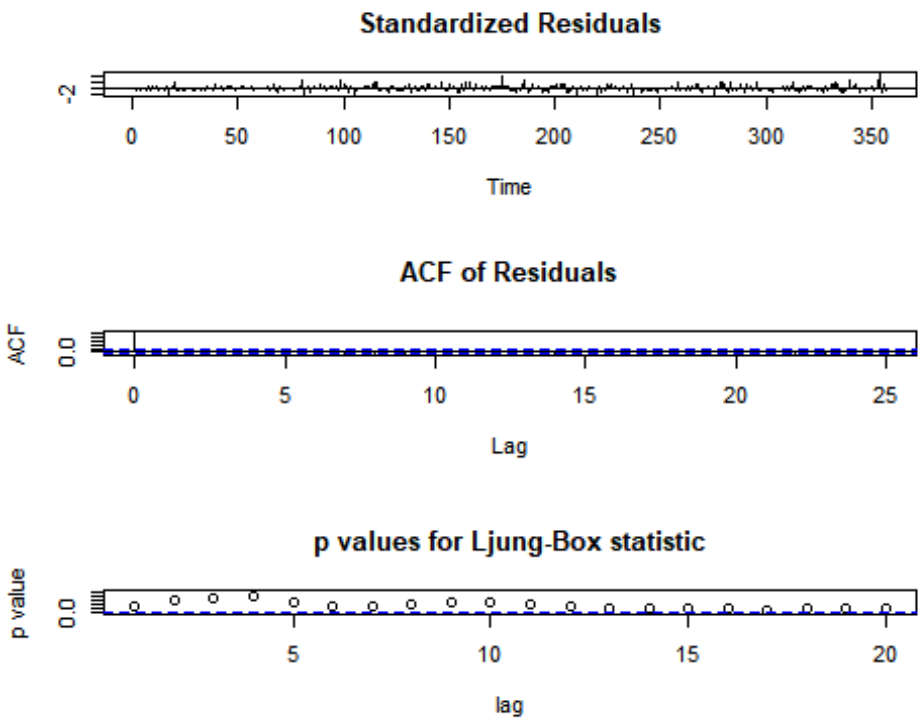
```
AIC(spam.auto)
```

```
## [1] 2491.98
```

```
#AIC =  2491.98
BIC(spam.auto)
```

```
## [1] 2503.614
```

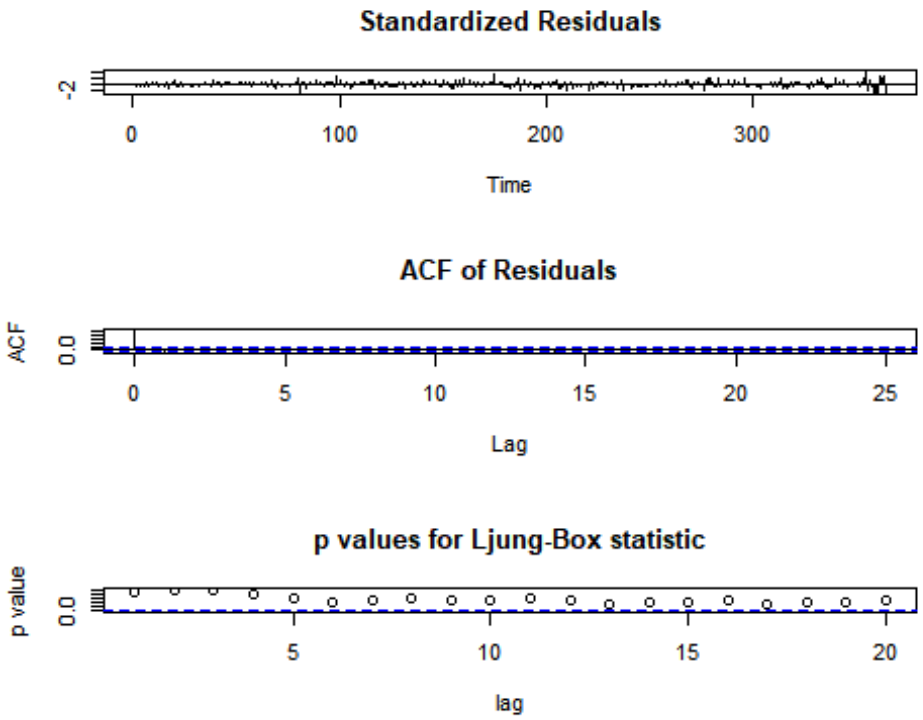
```
#BIC = 2503.614
tsdiag(spam.auto,gof.lag=20)
```




```
#Diagnostics for Ljung-Box Statistic: Model is very adequate because of high
p-values for Ljung Box statistic, accept null hypothesis.
```

```r
#On whole spam data set

#Auto generated model on whole data set
spam.auto.whole <- auto.arima(spam.ts, trace=TRUE)

##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2) with drift         : 2600.104
##  ARIMA(0,1,0) with drift         : 2767.792
##  ARIMA(1,1,0) with drift         : 2701.632
##  ARIMA(0,1,1) with drift         : 2601.85
##  ARIMA(0,1,0)                    : 2765.772
##  ARIMA(1,1,2) with drift         : 2598.321
##  ARIMA(1,1,1) with drift         : 2596.691
##  ARIMA(1,1,1)                    : 2595.899
##  ARIMA(0,1,1)                    : 2600.417
##  ARIMA(2,1,1)                    : 2597.334
##  ARIMA(1,1,0)                    : 2699.601
##  ARIMA(1,1,2)                    : 2597.454
##  ARIMA(2,1,2)                    : 2596.24
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(1,1,1)                    : 2600.173
##
##  Best model: ARIMA(1,1,1)

summary(spam.auto.whole)

## Series: spam.ts
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.161  -0.9449
## s.e.  0.056   0.0180
##
## sigma^2 estimated as 74.34:  log likelihood=-1297.05
## AIC=2600.11   AICc=2600.17   BIC=2611.79
##
## Training set error measures:
##                      ME     RMSE      MAE  MPE MAPE      MASE          ACF1
## Training set 0.4126606 8.586375 6.409686 -Inf  Inf 0.7810392 -0.004574539

AIC(spam.auto.whole)

## [1] 2600.107

#AIC = 2600.107
BIC(spam.auto.whole)
```

```
## [1] 2611.79

#BIC = 2611.79
tsdiag(spam.auto.whole,gof.lag=20)
```




```
#Diagnostics for Ljung-Box Statistic: Model is adequate because of high p-values for Ljung Box statistic, accept null hypothesis.

# Estimated ARMA on whole with p=1 and q=1
spam.arma11.whole <- arima(spam.ts, order=c(1,0,1))
summary(spam.arma11.whole)

##
## Call:
## arima(x = spam.ts, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.6179  -0.3797    27.3612
## s.e.  0.1742   0.2096     0.7398
##
## sigma^2 estimated as 76.02:  log likelihood = -1304.78,  aic = 2617.55
##
## Training set error measures:
##                       ME     RMSE      MAE  MPE MAPE      MASE        ACF1
## Training set 0.008751358 8.718725 6.535412 -Inf  Inf 0.7963594 0.007283154
```

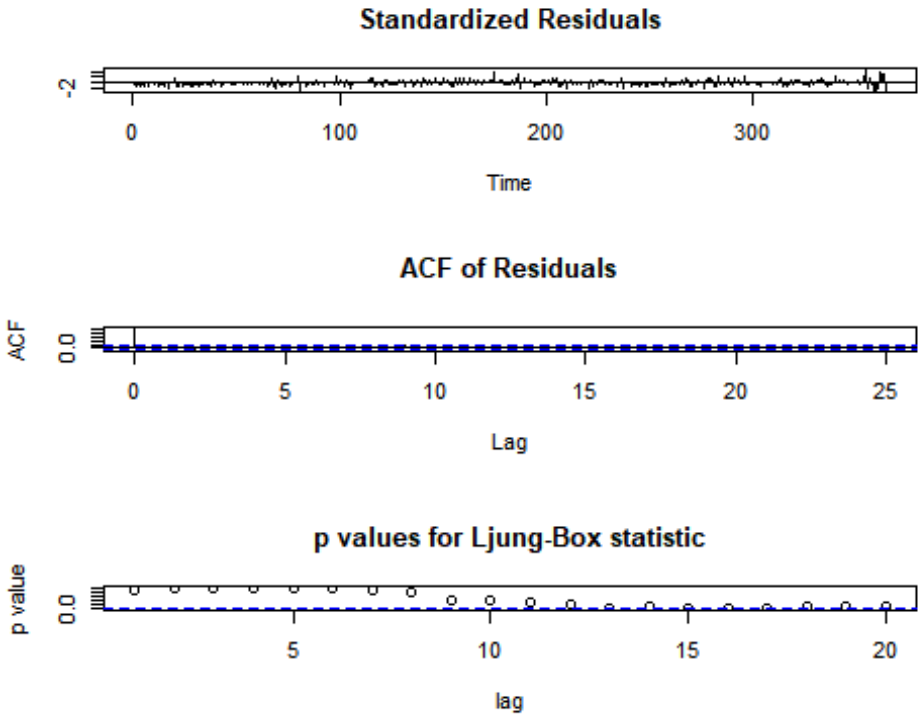
```r
AIC(spam.arma11.whole)
```

```
## [1] 2617.553
```

```r
#AIC = 2617.553
BIC(spam.arma11.whole)
```

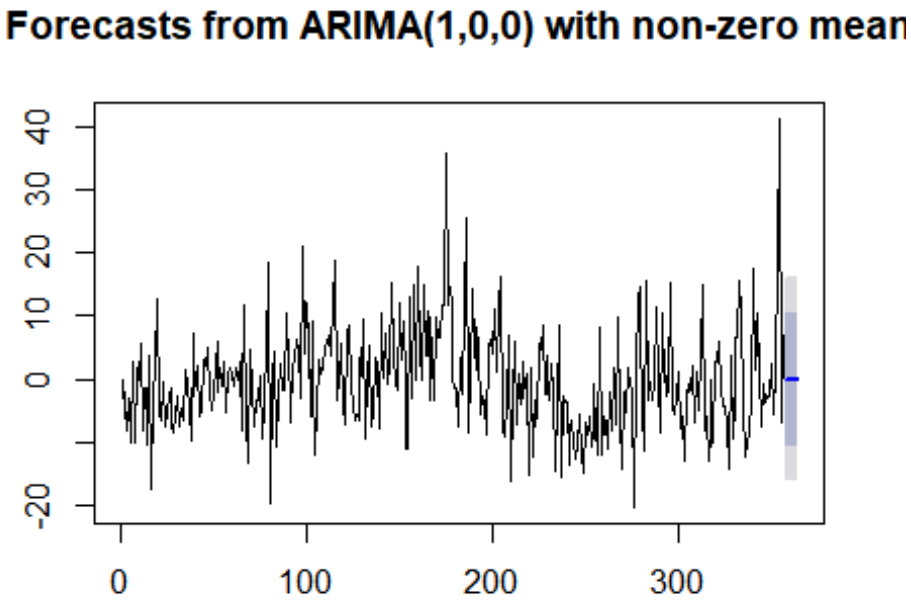
```
## [1] 2633.142
```

```r
#BIC = 2633.142
tsdiag(spam.arma11.whole,gof.lag=20)
```

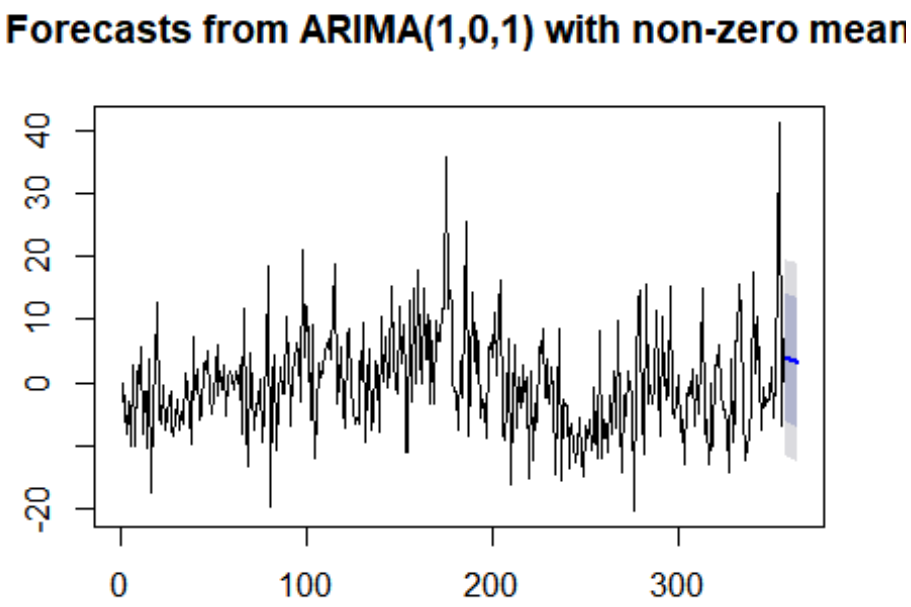



```r
#Diagnostics for Ljung-Box Statistic: Model is roughly not adequate after lag of 12 equates to low p-values.


#Based on AIC, BIC, and Diagnostics, spam.arima111 with ARIMA(1,1,1), which performs the best because of low AIC, BIC, and best diagnostics.

#FORECASTING

# Prediction performance

#Forecast the next 7 days of spam with the forecast option for each model
spam.ar1.forecast<-forecast(spam.ar1,h=7)
spam.arma11.forecast<-forecast(spam.arma11,h=7)
spam.arima111.forecast<-forecast(spam.arima111,h=7)
```

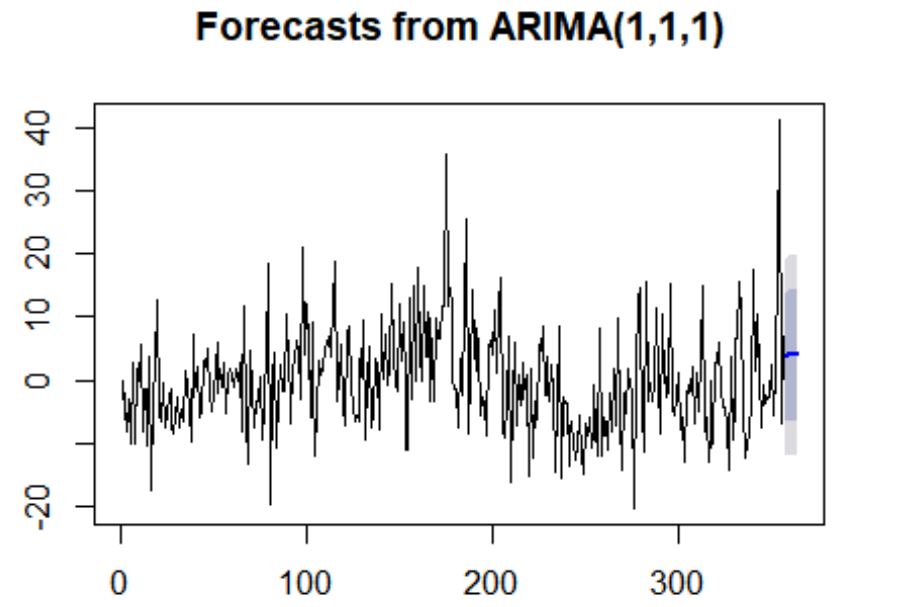
```
spam.auto.forecast<-forecast(spam.auto,h=7)
spam.auto.whole.forecast<-forecast(spam.auto.whole,h=7)
spam.arma11.whole.forecast<-forecast(spam.arma11.whole,h=7)

#View plots of the forecasts of each model
plot(spam.ar1.forecast)
```



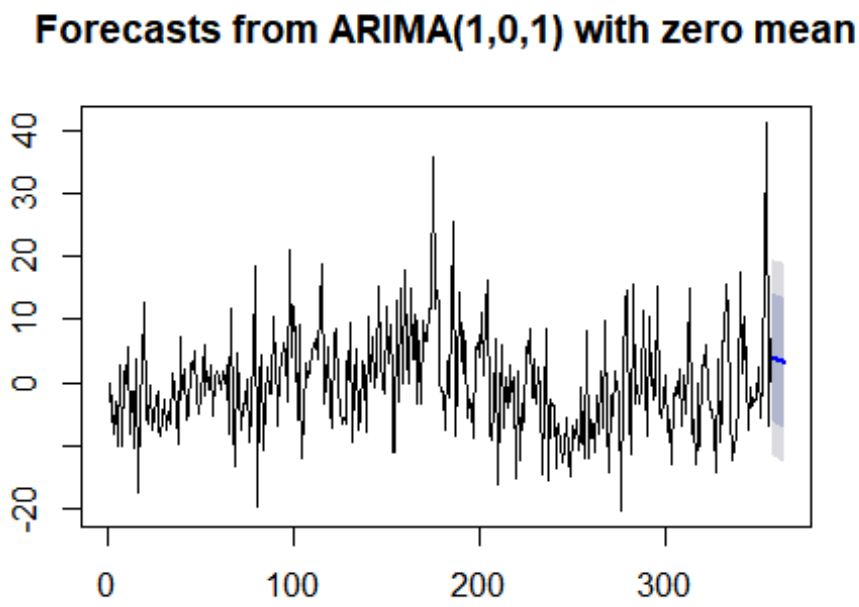

```
plot(spam.arma11.forecast)
```



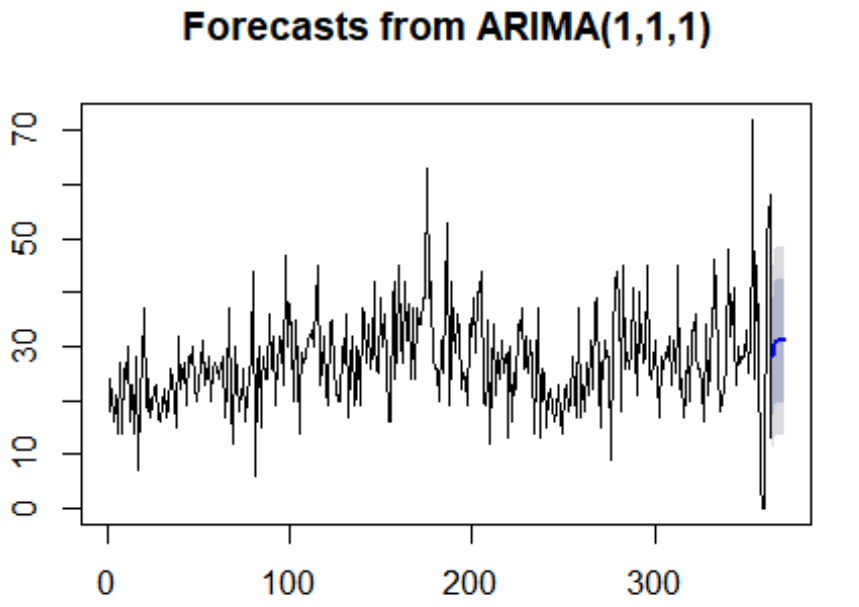

```
plot(spam.arima111.forecast)
```



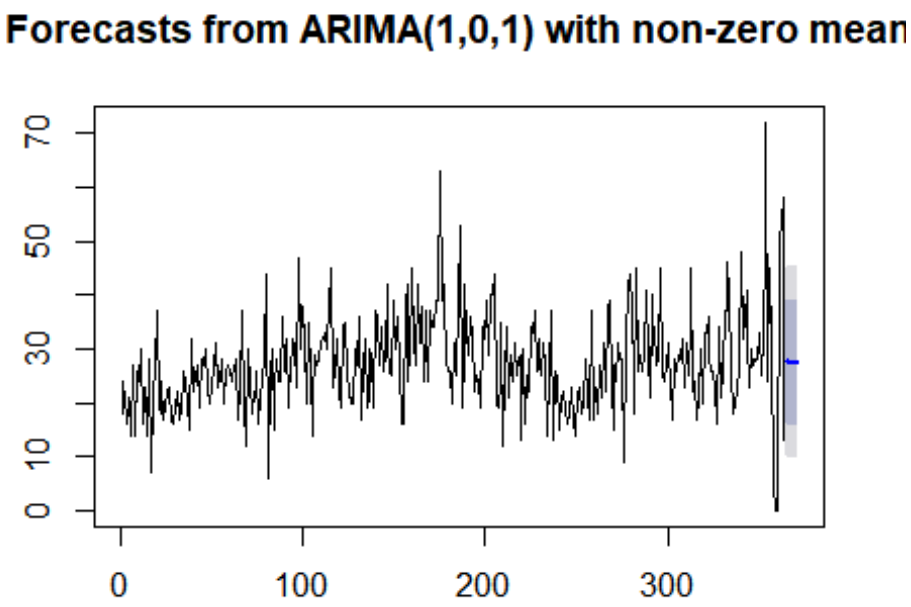

```
plot(spam.auto.forecast)
```




```
plot(spam.auto.whole.forecast)
```




```
plot(spam.arma11.whole.forecast)
```




```
#Seems that spam.arima111 may forecast the best


# Create test set from spam data set with last 7 days
# The next week or the test period in days
next.week.time<-c((length(spam.ts)-6):(length(spam.ts)))
# Create the test data frame
next.week<-data.frame(time.spam = next.week.time, count =
spam.ts[next.week.time])
# Create the actual time series for the test period
next.week.ts <- spam.ts[next.week.time]
next.week.ts<-ts(next.week$count)

#Create each of the predictions for the next week for each of the 6 models
next.week.prediction.ar1<-
predict(spam.trend,newdata=next.week)+forecast(spam.ar1,h=7)$mean
next.week.prediction.arma11<-
predict(spam.trend,newdata=next.week)+forecast(spam.arma11,h=7)$mean
next.week.prediction.arima111<-
predict(spam.trend,newdata=next.week)+forecast(spam.arima111,h=7)$mean
next.week.prediction.auto<-
predict(spam.trend,newdata=next.week)+forecast(spam.auto,h=7)$mean
next.week.prediction.auto.whole<-
predict(spam.trend,newdata=next.week)+forecast(spam.auto.whole,h=7)$mean
next.week.prediction.arma11.whole<-
predict(spam.trend,newdata=next.week)+forecast(spam.arma11.whole,h=7)$mean
```

```r
# MSE of predictions
mean((next.week.prediction.ar1-next.week$count)^2) #646.7569
```

```
## [1] 646.7569
```

```r
mean((next.week.prediction.arma11-next.week$count)^2) # 704.0463
```

```
## [1] 704.0463
```

```r
mean((next.week.prediction.arima111-next.week$count)^2) # 699.8328
```

```
## [1] 699.8328
```

```r
mean((next.week.prediction.auto-next.week$count)^2) #  703.4861
```

```
## [1] 703.4861
```

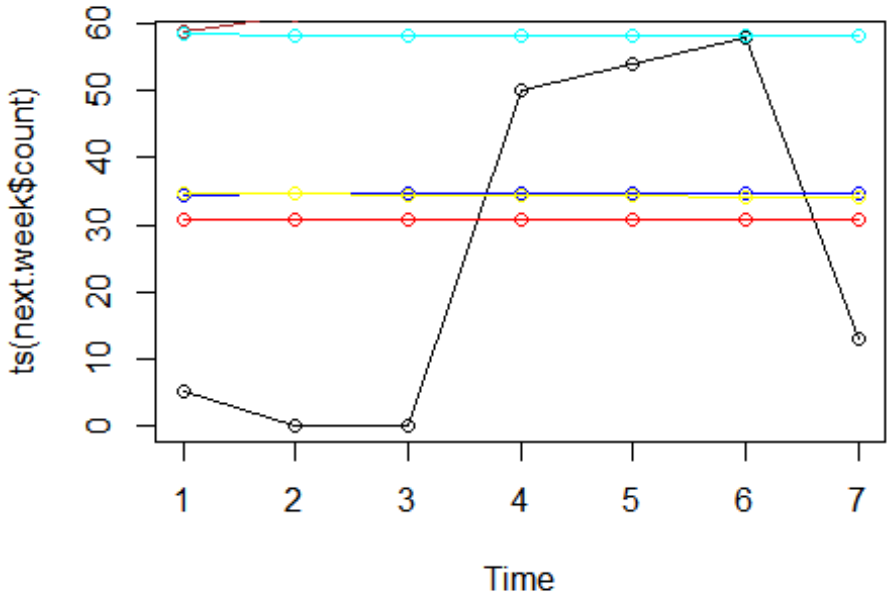
```r
mean((next.week.prediction.auto.whole-next.week$count)^2) # 1870.459
```

```
## [1] 1870.459
```

```r
mean((next.week.prediction.arma11.whole-next.week$count)^2) #1689.32
```

```
## [1] 1689.32
```

```r
#The AR1 model has the lowest MSE value with 646.7569, with the arima111
model having the second lowest with 699.8328, and the third lowest being the
spam.auto model that was generated.

# Using plots to compare the predicted value and real values for the last
week of data
plot(ts(next.week$count),type='o')
lines(ts(next.week.prediction.ar1),col='red',type='o')
lines(ts(next.week.prediction.arma11),col='green',type='o')
lines(ts(next.week.prediction.arima111),col='blue',type='o')
lines(ts(next.week.prediction.auto),col='yellow',type='o')
lines(ts(next.week.prediction.auto.whole),col='brown',type='o')
lines(ts(next.week.prediction.arma11.whole),col='cyan',type='o')
```




```
#It is clear from the graph that the blue and yellow models perform the best,
as they are very similar to the predicted values. The models which used the
entire spam time series data were not very predictive at all.

#Problems with models: Some models had low p-values under the Ljung-Box
statistic after a certain amount of lags.
#How adjust models: Since this is not a problem for most of the models, it
would most likely be best to discard these models, as they are not the most
feasible models in comparison to the other models.
```

d)Which model do you recommend and why?

For the spam time series data, the estimated ARIMA model with (1,1,1) is the recommended model. This is because the model performed the best in terms of the AIC and BIC with 2489.917 and 2501.541 respectively. Also, it performed well in terms of the model diagnostics with the model being adequate. Also, the MSE was quite low with 699.8328 and it was the best model for forecasting predicted vs actual values. The AR model, while it had a low MSE, was not adequate due to the Ljung Box Statistic. The auto generated model of ARIMA (1,0,1) was fairly close in terms of AIC, BIC, and MSE, but did not perform as well as the ARIMA (1,1,1) model. The models that used the whole spam time series data set did not perform well over with high MSE, AIC, and BIC values. Therefore, the ARIMA (1,1,1) model proves to be the best in terms of performance statistics.

3. Integrated Filter Design

(a) Use Baye's rule to derive the equation for an integrated spam filter that combines your static and time series filters. Define all of the parameter used.

*Included on Attached Word Document