Analysis of Spam Emails via Trees

Jhonasttan Regalado, Yehui He, Yisong Tao, Regan Yee, Connie Zhang

November 21, 2016

EDA

```
## Summary functions of Spam Emails:
dim(spam)
```

```
## [1] 4601   58
```

```
colnames(spam)
```

```
##  [1] "make"              "address"           "all"
##  [4] "num3d"             "our"               "over"
##  [7] "remove"            "internet"          "order"
## [10] "mail"              "receive"           "will"
## [13] "people"            "report"            "addresses"
## [16] "free"              "business"          "email"
## [19] "you"               "credit"            "your"
## [22] "font"              "num000"            "money"
## [25] "hp"                "hpl"               "george"
## [28] "num650"            "lab"               "labs"
## [31] "telnet"            "num857"            "data"
## [34] "num415"            "num85"             "technology"
## [37] "num1999"           "parts"             "pm"
## [40] "direct"            "cs"                "meeting"
## [43] "original"          "project"           "re"
## [46] "edu"               "table"             "conference"
## [49] "charSemicolon"     "charRoundbracket"  "charSquarebracket"
## [52] "charExclamation"   "charDollar"        "charHash"
## [55] "capitalAve"        "capitalLong"       "capitalTotal"
## [58] "type"
```

```
summary(spam)
```

```
##       make           address             all              num3d
##  Min.   :0.0000   Min.   : 0.000   Min.   :0.0000   Min.   : 0.00000
##  1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.: 0.00000
##  Median :0.0000   Median : 0.000   Median :0.0000   Median : 0.00000
##  Mean   :0.1046   Mean   : 0.213   Mean   :0.2807   Mean   : 0.06542
##  3rd Qu.:0.0000   3rd Qu.: 0.000   3rd Qu.:0.4200   3rd Qu.: 0.00000
##  Max.   :4.5400   Max.   :14.280   Max.   :5.1000   Max.   :42.81000
##       our              over             remove           internet
##  Min.   : 0.0000   Min.   :0.0000   Min.   :0.0000   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median :0.0000   Median :0.0000   Median : 0.0000
##  Mean   : 0.3122   Mean   :0.0959   Mean   :0.1142   Mean   : 0.1053
##  3rd Qu.: 0.3800   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.: 0.0000
##  Max.   :10.0000   Max.   :5.8800   Max.   :7.2700   Max.   :11.1100
##      order              mail            receive             will
##  Min.   :0.00000   Min.   : 0.0000   Min.   :0.00000   Min.   :0.0000
```

```
##  1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median : 0.0000   Median :0.00000   Median :0.1000
##  Mean   :0.09007   Mean   : 0.2394   Mean   :0.05982   Mean   :0.5417
##  3rd Qu.:0.00000   3rd Qu.: 0.1600   3rd Qu.:0.00000   3rd Qu.:0.8000
##  Max.   :5.26000   Max.   :18.1800   Max.   :2.61000   Max.   :9.6700
##      people            report          addresses           free
##  Min.   :0.00000   Min.   : 0.00000   Min.   :0.0000   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.: 0.00000   1st Qu.:0.0000   1st Qu.: 0.0000
##  Median :0.00000   Median : 0.00000   Median :0.0000   Median : 0.0000
##  Mean   :0.09393   Mean   : 0.05863   Mean   :0.0492   Mean   : 0.2488
##  3rd Qu.:0.00000   3rd Qu.: 0.00000   3rd Qu.:0.0000   3rd Qu.: 0.1000
##  Max.   :5.55000   Max.   :10.00000   Max.   :4.4100   Max.   :20.0000
##     business          email             you             credit
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   : 0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.: 0.00000
##  Median :0.0000   Median :0.0000   Median : 1.310   Median : 0.00000
##  Mean   :0.1426   Mean   :0.1847   Mean   : 1.662   Mean   : 0.08558
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.: 2.640   3rd Qu.: 0.00000
##  Max.   :7.1400   Max.   :9.0900   Max.   :18.750   Max.   :18.18000
##       your              font            num000           money
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :0.0000   Min.   : 0.00000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.: 0.00000
##  Median : 0.2200   Median : 0.0000   Median :0.0000   Median : 0.00000
##  Mean   : 0.8098   Mean   : 0.1212   Mean   :0.1016   Mean   : 0.09427
##  3rd Qu.: 1.2700   3rd Qu.: 0.0000   3rd Qu.:0.0000   3rd Qu.: 0.00000
##  Max.   :11.1100   Max.   :17.1000   Max.   :5.4500   Max.   :12.50000
##        hp               hpl              george           num650
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   :0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median :0.0000
##  Mean   : 0.5495   Mean   : 0.2654   Mean   : 0.7673   Mean   :0.1248
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:0.0000
##  Max.   :20.8300   Max.   :16.6600   Max.   :33.3300   Max.   :9.0900
##       lab               labs             telnet            num857
##  Min.   : 0.00000   Min.   :0.0000   Min.   : 0.00000   Min.   :0.00000
##  1st Qu.: 0.00000   1st Qu.:0.0000   1st Qu.: 0.00000   1st Qu.:0.00000
##  Median : 0.00000   Median :0.0000   Median : 0.00000   Median :0.00000
##  Mean   : 0.09892   Mean   :0.1029   Mean   : 0.06475   Mean   :0.04705
##  3rd Qu.: 0.00000   3rd Qu.:0.0000   3rd Qu.: 0.00000   3rd Qu.:0.00000
##  Max.   :14.28000   Max.   :5.8800   Max.   :12.50000   Max.   :4.76000
##       data              num415            num85           technology
##  Min.   : 0.00000   Min.   :0.00000   Min.   : 0.0000   Min.   :0.00000
##  1st Qu.: 0.00000   1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.00000
##  Median : 0.00000   Median :0.00000   Median : 0.0000   Median :0.00000
##  Mean   : 0.09723   Mean   :0.04784   Mean   : 0.1054   Mean   :0.09748
##  3rd Qu.: 0.00000   3rd Qu.:0.00000   3rd Qu.: 0.0000   3rd Qu.:0.00000
##  Max.   :18.18000   Max.   :4.76000   Max.   :20.0000   Max.   :7.69000
##     num1999           parts              pm              direct
##  Min.   :0.000   Min.   :0.0000   Min.   : 0.00000   Min.   :0.00000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.: 0.00000   1st Qu.:0.00000
##  Median :0.000   Median :0.0000   Median : 0.00000   Median :0.00000
##  Mean   :0.137   Mean   :0.0132   Mean   : 0.07863   Mean   :0.06483
##  3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.: 0.00000   3rd Qu.:0.00000
##  Max.   :6.890   Max.   :8.3300   Max.   :11.11000   Max.   :4.76000
```

```
##        cs               meeting           original         project
##  Min.   :0.00000   Min.   : 0.0000   Min.   :0.0000   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.: 0.0000
##  Median :0.00000   Median : 0.0000   Median :0.0000   Median : 0.0000
##  Mean   :0.04367   Mean   : 0.1323   Mean   :0.0461   Mean   : 0.0792
##  3rd Qu.:0.00000   3rd Qu.: 0.0000   3rd Qu.:0.0000   3rd Qu.: 0.0000
##  Max.   :7.14000   Max.   :14.2800   Max.   :3.5700   Max.   :20.0000
##        re                edu              table            conference
##  Min.   : 0.0000   Min.   : 0.0000   Min.   :0.000000   Min.   : 0.00000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.000000   1st Qu.: 0.00000
##  Median : 0.0000   Median : 0.0000   Median :0.000000   Median : 0.00000
##  Mean   : 0.3012   Mean   : 0.1798   Mean   :0.005444   Mean   : 0.03187
##  3rd Qu.: 0.1100   3rd Qu.: 0.0000   3rd Qu.:0.000000   3rd Qu.: 0.00000
##  Max.   :21.4200   Max.   :22.0500   Max.   :2.170000   Max.   :10.00000
##  charSemicolon     charRoundbracket charSquarebracket charExclamation
##  Min.   :0.00000   Min.   :0.000    Min.   :0.00000   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.:0.000    1st Qu.:0.00000   1st Qu.: 0.0000
##  Median :0.00000   Median :0.065    Median :0.00000   Median : 0.0000
##  Mean   :0.03857   Mean   :0.139    Mean   :0.01698   Mean   : 0.2691
##  3rd Qu.:0.00000   3rd Qu.:0.188    3rd Qu.:0.00000   3rd Qu.: 0.3150
##  Max.   :4.38500   Max.   :9.752    Max.   :4.08100   Max.   :32.4780
##    charDollar         charHash          capitalAve        capitalLong
##  Min.   :0.00000   Min.   : 0.00000   Min.   :   1.000   Min.   :   1.00
##  1st Qu.:0.00000   1st Qu.: 0.00000   1st Qu.:   1.588   1st Qu.:   6.00
##  Median :0.00000   Median : 0.00000   Median :   2.276   Median :  15.00
##  Mean   :0.07581   Mean   : 0.04424   Mean   :   5.191   Mean   :  52.17
##  3rd Qu.:0.05200   3rd Qu.: 0.00000   3rd Qu.:   3.706   3rd Qu.:  43.00
##  Max.   :6.00300   Max.   :19.82900   Max.   :1102.500   Max.   :9989.00
##   capitalTotal          type
##  Min.   :    1.0   nonspam:2788
##  1st Qu.:   35.0   spam   :1813
##  Median :   95.0
##  Mean   :  283.3
##  3rd Qu.:  266.0
##  Max.   :15841.0
```

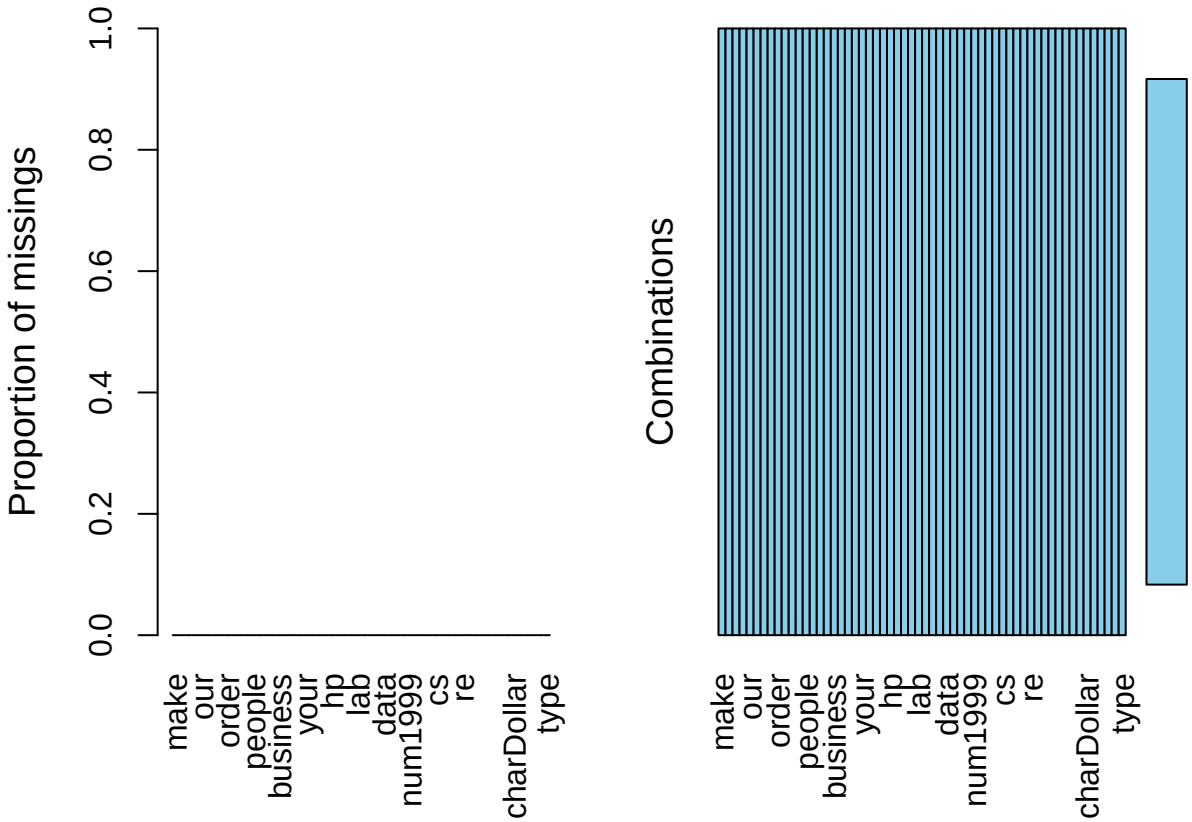
```
head(spam)
```

```
##   make address  all num3d  our over remove internet order mail receive
## 1 0.00    0.64 0.64     0 0.32 0.00   0.00     0.00  0.00 0.00    0.00
## 2 0.21    0.28 0.50     0 0.14 0.28   0.21     0.07  0.00 0.94    0.21
## 3 0.06    0.00 0.71     0 1.23 0.19   0.19     0.12  0.64 0.25    0.38
## 4 0.00    0.00 0.00     0 0.63 0.00   0.31     0.63  0.31 0.63    0.31
## 5 0.00    0.00 0.00     0 0.63 0.00   0.31     0.63  0.31 0.63    0.31
## 6 0.00    0.00 0.00     0 1.85 0.00   0.00     1.85  0.00 0.00    0.00
##   will people report addresses free business email  you credit your font
## 1 0.64   0.00   0.00      0.00 0.32     0.00  1.29 1.93   0.00 0.96    0
## 2 0.79   0.65   0.21      0.14 0.14     0.07  0.28 3.47   0.00 1.59    0
## 3 0.45   0.12   0.00      1.75 0.06     0.06  1.03 1.36   0.32 0.51    0
## 4 0.31   0.31   0.00      0.00 0.31     0.00  0.00 3.18   0.00 0.31    0
## 5 0.31   0.31   0.00      0.00 0.31     0.00  0.00 3.18   0.00 0.31    0
## 6 0.00   0.00   0.00      0.00 0.00     0.00  0.00 0.00   0.00 0.00    0
##   num000 money hp hpl george num650 lab labs telnet num857 data num415
## 1   0.00  0.00  0   0      0      0   0    0      0      0    0      0
## 2   0.43  0.43  0   0      0      0   0    0      0      0    0      0
```

```
## 3   1.16  0.06  0   0      0      0   0    0      0      0    0      0
## 4   0.00  0.00  0   0      0      0   0    0      0      0    0      0
## 5   0.00  0.00  0   0      0      0   0    0      0      0    0      0
## 6   0.00  0.00  0   0      0      0   0    0      0      0    0      0
##   num85 technology num1999 parts pm direct cs meeting original project
## 1     0          0    0.00     0  0   0.00  0       0     0.00       0
## 2     0          0    0.07     0  0   0.00  0       0     0.00       0
## 3     0          0    0.00     0  0   0.06  0       0     0.12       0
## 4     0          0    0.00     0  0   0.00  0       0     0.00       0
## 5     0          0    0.00     0  0   0.00  0       0     0.00       0
## 6     0          0    0.00     0  0   0.00  0       0     0.00       0
##     re  edu table conference charSemicolon charRoundbracket
## 1 0.00 0.00     0          0          0.00            0.000
## 2 0.00 0.00     0          0          0.00            0.132
## 3 0.06 0.06     0          0          0.01            0.143
## 4 0.00 0.00     0          0          0.00            0.137
## 5 0.00 0.00     0          0          0.00            0.135
## 6 0.00 0.00     0          0          0.00            0.223
##   charSquarebracket charExclamation charDollar charHash capitalAve
## 1                 0           0.778      0.000    0.000      3.756
## 2                 0           0.372      0.180    0.048      5.114
## 3                 0           0.276      0.184    0.010      9.821
## 4                 0           0.137      0.000    0.000      3.537
## 5                 0           0.135      0.000    0.000      3.537
## 6                 0           0.000      0.000    0.000      3.000
##   capitalLong capitalTotal type
## 1          61          278 spam
## 2         101         1028 spam
## 3         485         2259 spam
## 4          40          191 spam
## 5          40          191 spam
## 6          15           54 spam
```

SPAM dataset is complete (no NAs!!)

```
aggr(spam) ## no missingness
```




```
library(kernlab)
library(dplyr)

data(spam)
spam_features = spam %>% select (-type)
spam_mean = sapply(spam_features, mean)

spam_mean_bytype = spam %>% group_by(type) %>% summarise_each(funs(mean))

spam_mean_bytype = as.data.frame(spam_mean_bytype)
difference = data.frame()
difference[1,1] = 'difference'


for (i in(2:ncol(spam_mean_bytype))) {
  difference[1,i] = spam_mean_bytype[1,i] - spam_mean_bytype[2,i]
}

colnames = colnames(spam_mean_bytype)
names(difference) = colnames

NonspamVSspam = rbind(spam_mean_bytype, difference)


DiffOnMean = t(NonspamVSspam)
DiffOnMean= as.data.frame(DiffOnMean)
colnames(DiffOnMean) = c('nonSpam','Spam','Mean Difference')
DiffOnMean =DiffOnMean[2:nrow(DiffOnMean),]
```

```
DiffOnMean
```

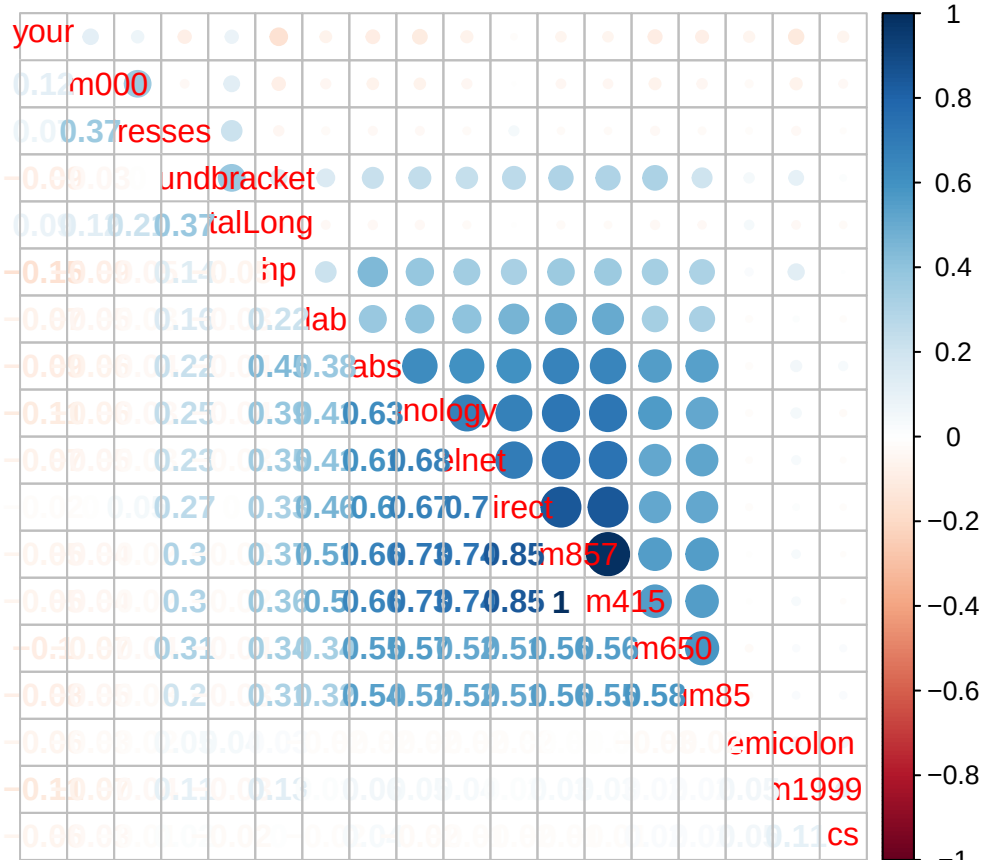
```
##                          nonSpam          Spam Mean Difference
## make                  0.07347920    0.15233867     -0.07885947
## address               0.24446557    0.16464975      0.07981581
## all                   0.2005811     0.4037948      -0.2032138
## num3d              0.0008859397  0.1646718147   -0.1637858749
## our                   0.1810402     0.5139548      -0.3329146
## over                  0.04454448    0.17487590     -0.13033142
## remove                0.00938307    0.27540541     -0.26602234
## internet              0.03841463    0.20814120     -0.16972657
## order                 0.03804878    0.17006067     -0.13201189
## mail                  0.1671700     0.3505074      -0.1833374
## receive               0.02171090    0.11843354     -0.09672263
## will                  0.53632353    0.54997242     -0.01364889
## people                0.06166428    0.14354661     -0.08188233
## report                0.04240316    0.08357419     -0.04117103
## addresses             0.008317791   0.112079426    -0.103761636
## free                  0.0735868     0.5183618      -0.4447750
## business              0.04834648    0.28750689     -0.23916041
## email                 0.09729197    0.31922780     -0.22193583
## you                   1.2703407     2.2645394      -0.9941987
## credit                0.00757891    0.20552124     -0.19794233
## your                  0.4387016     1.3803696      -0.9416680
## font                  0.04522597    0.23803640     -0.19281044
## num000                0.007087518   0.247054606    -0.239967088
## money                 0.01713773    0.21287921     -0.19574147
## hp                    0.89547346    0.01747932      0.87799414
## hpl                   0.431994261   0.009172642     0.422821619
## george                1.265265423   0.001549917     1.263715506
## num650                0.19380560    0.01879757      0.17500802
## lab                0.1627941176  0.0006839493    0.1621101684
## labs                  0.165853659   0.005968009     0.159885650
## telnet                0.106032999   0.001274131     0.104758867
## num857             0.0773063128  0.0005184777    0.0767878351
## data                  0.1509864     0.0145615       0.1364249
## num415                0.077786944   0.001776062     0.076010882
## num85                 0.169454806   0.006927744     0.162527062
## technology            0.14167145    0.02951462      0.11215683
## num1999               0.19774390    0.04346939      0.15427451
## parts                 0.018723099   0.004710425     0.014012674
## pm                    0.12167862    0.01242692      0.10925171
## direct                0.08311693    0.03671815      0.04639878
## cs                 0.0720265423  0.0000551572    0.0719713851
## meeting               0.216807747   0.002443464     0.214364284
## original              0.070581062   0.008450083     0.062130979
## project               0.126635581   0.006243795     0.120391786
## re                    0.4157604     0.1250910       0.2906694
## edu                   0.28718436    0.01472697      0.27245739
## table                 0.008192253   0.001218974     0.006973278
## conference            0.051226686   0.002101489     0.049125197
## charSemicolon         0.05028085    0.02057308      0.02970776
## charRoundbracket      0.15857819    0.10897022      0.04960798
## charSquarebracket     0.022683644   0.008198566     0.014485078
```

```
## charExclamation       0.1099835     0.5137126      -0.4037291
## charDollar            0.01164849    0.17447821     -0.16282972
## charHash              0.02171306    0.07887700     -0.05716394
## capitalAve              2.377301      9.519165       -7.141864
## capitalLong             18.21449      104.39327      -86.17878
## capitalTotal            161.4709      470.6194       -309.1485
```

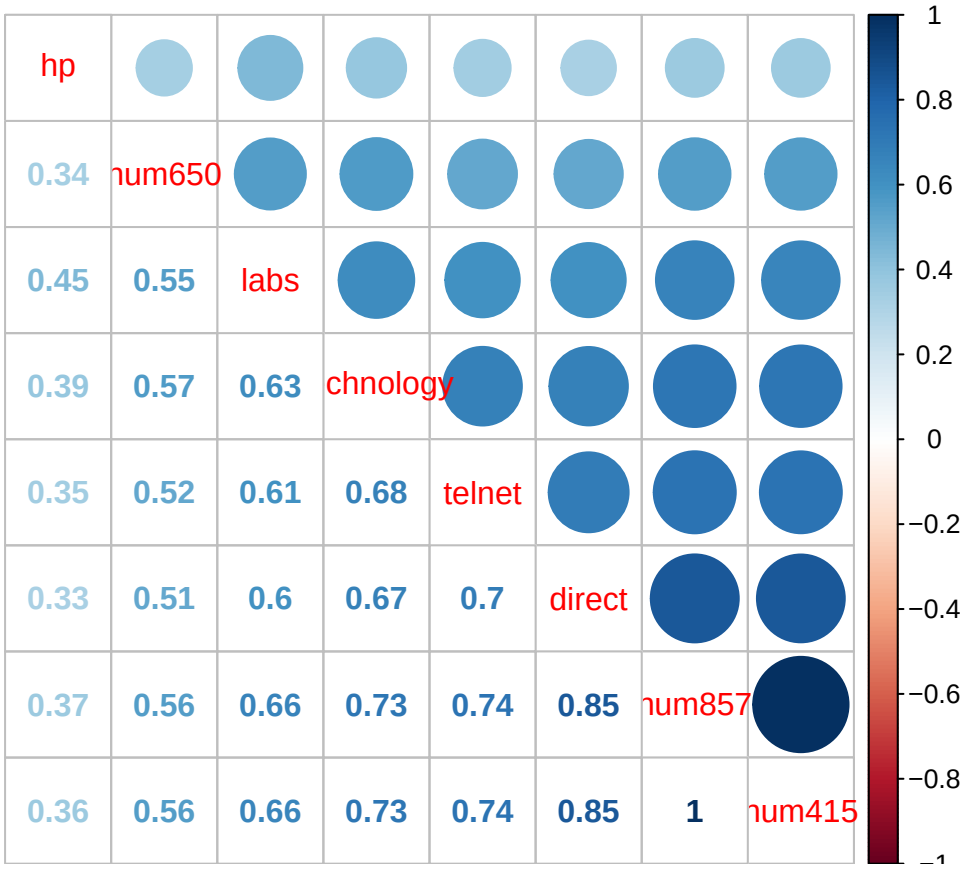
Correlation Analysis

Used the caret function 'findCorrelation' to identify highly correlated predictors. Goal was to determine which of these predictors would not be considered signficant for model prediction in Random Forest.

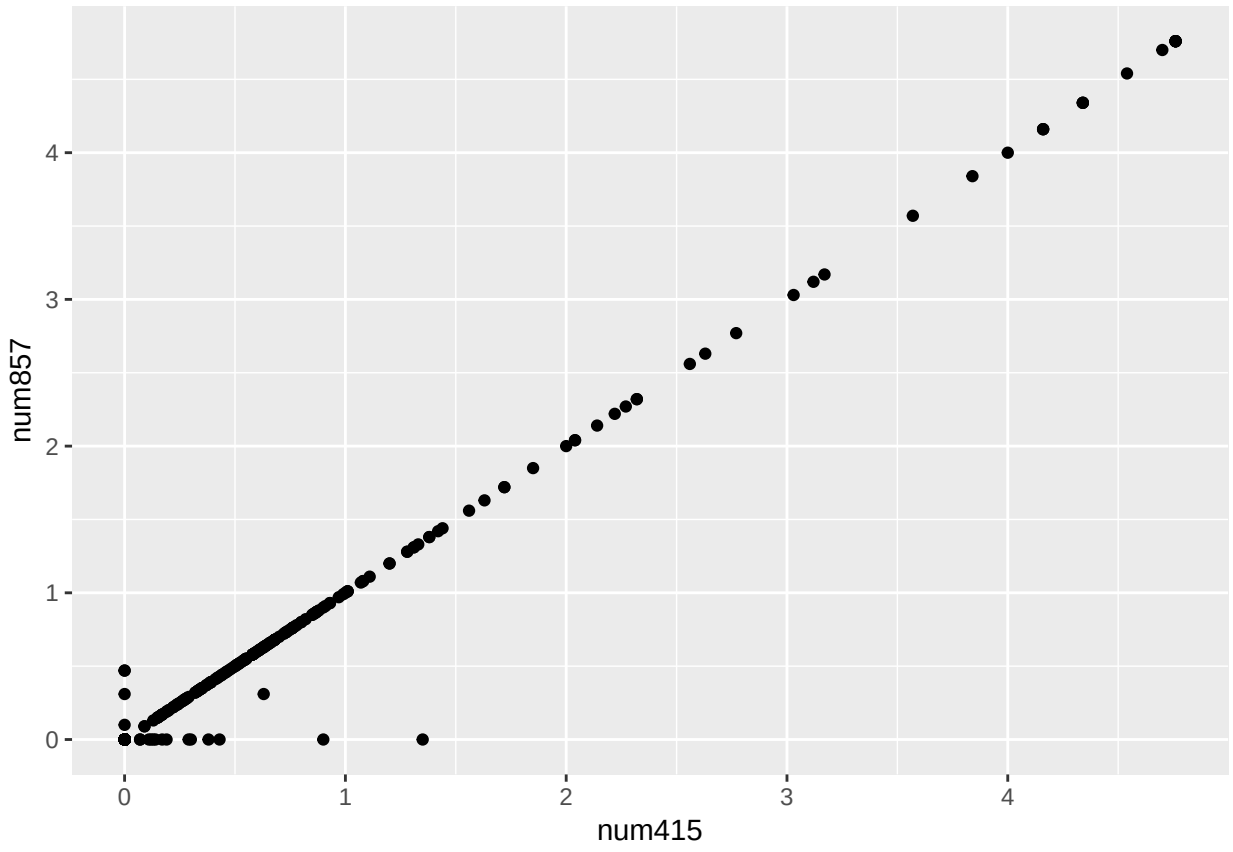
```
Cor<-cor(spam[,-58])
HhighCorr<-findCorrelation(Cor, cutoff=0.50)
LhighCorr<-findCorrelation(Cor, cutoff=0.25)

M2 <- cor(spam[,HhighCorr])
M3 <- cor(spam[,LhighCorr])
corrplot.mixed(M3, upper = "circle", order="hclust")
```




```
corrplot.mixed(M2, upper = "circle", order="hclust")
```

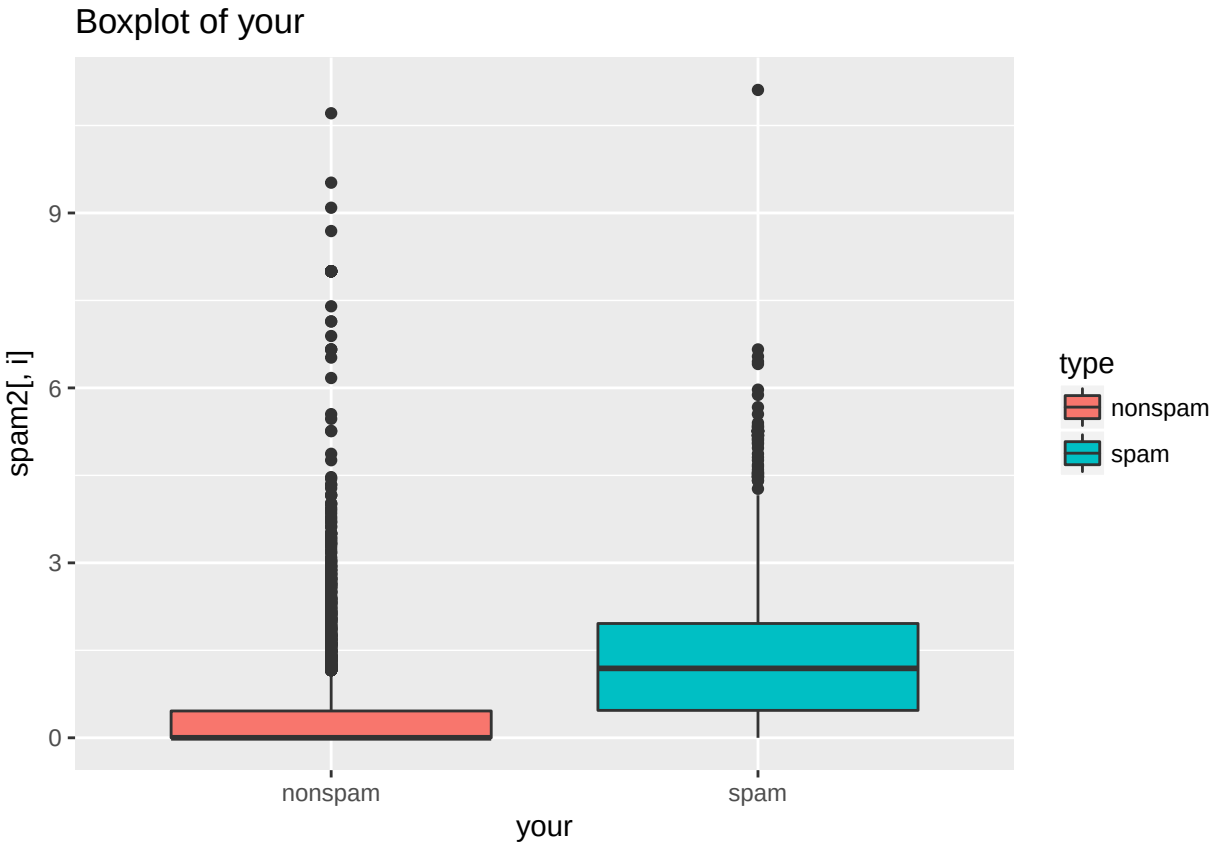


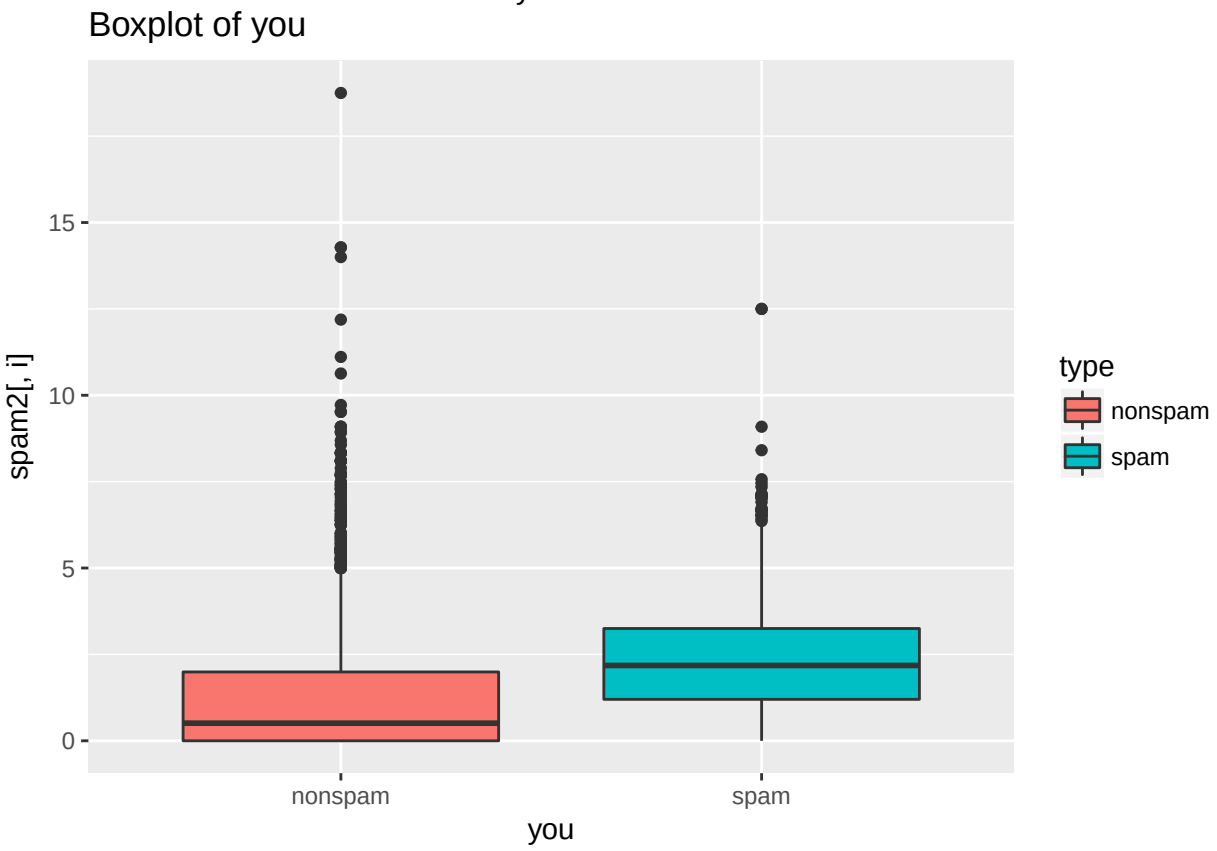

```
ggplot(data=spam, aes(num415, num857)) + geom_point()
```

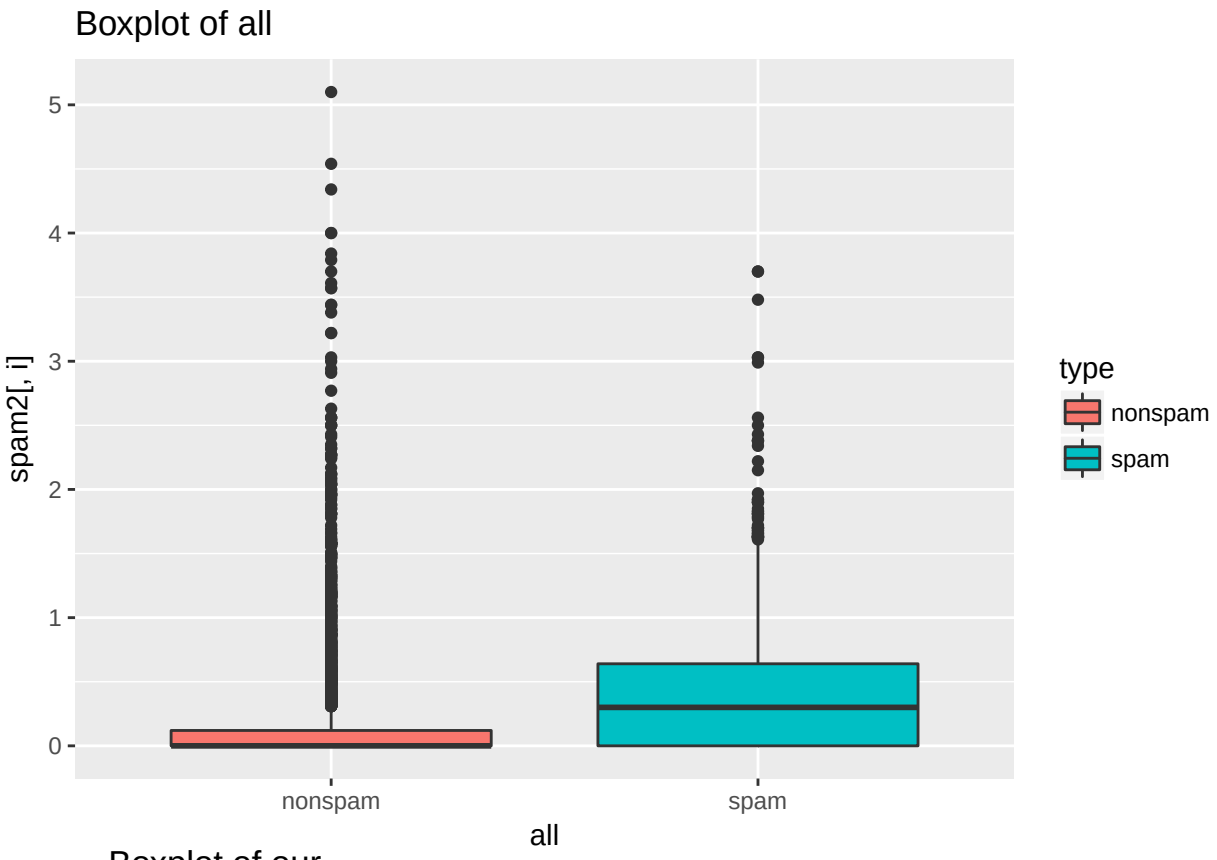


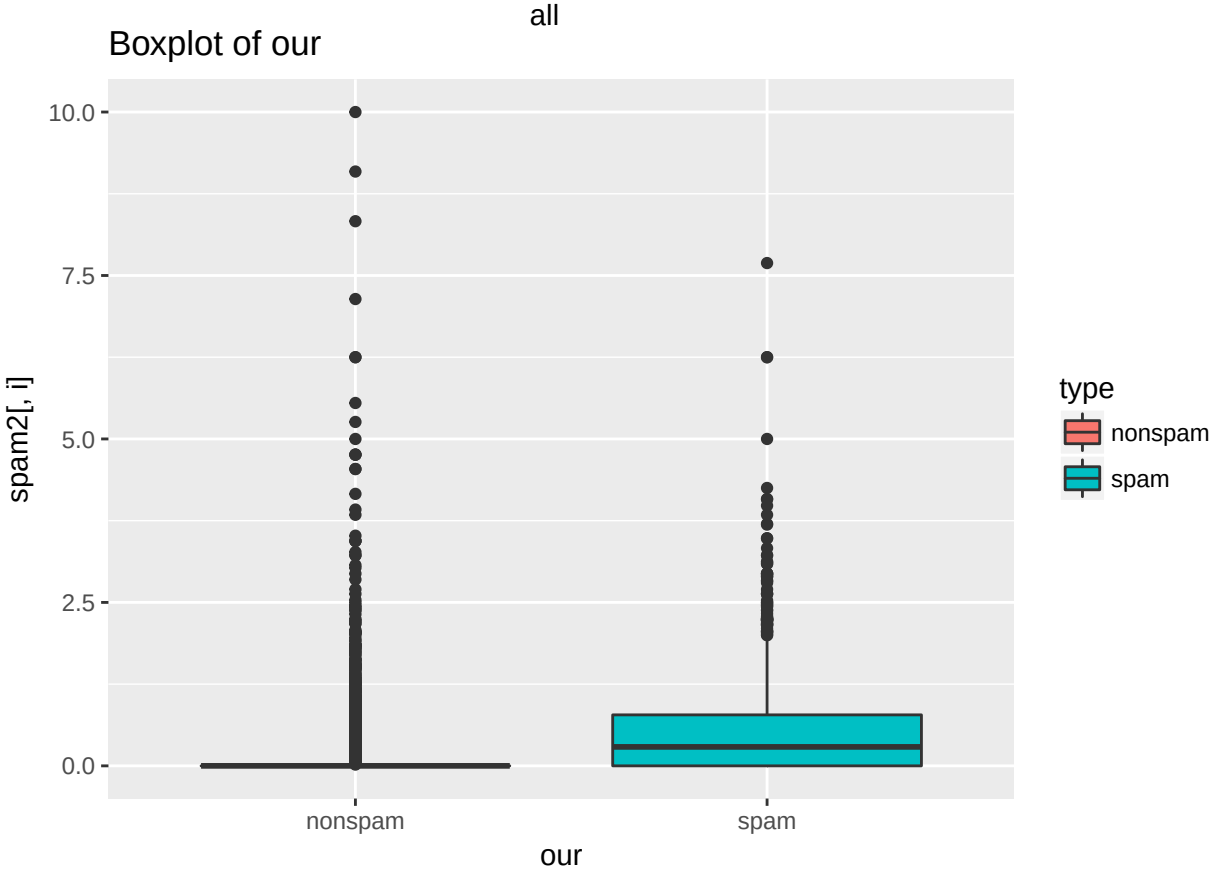
Box Plots

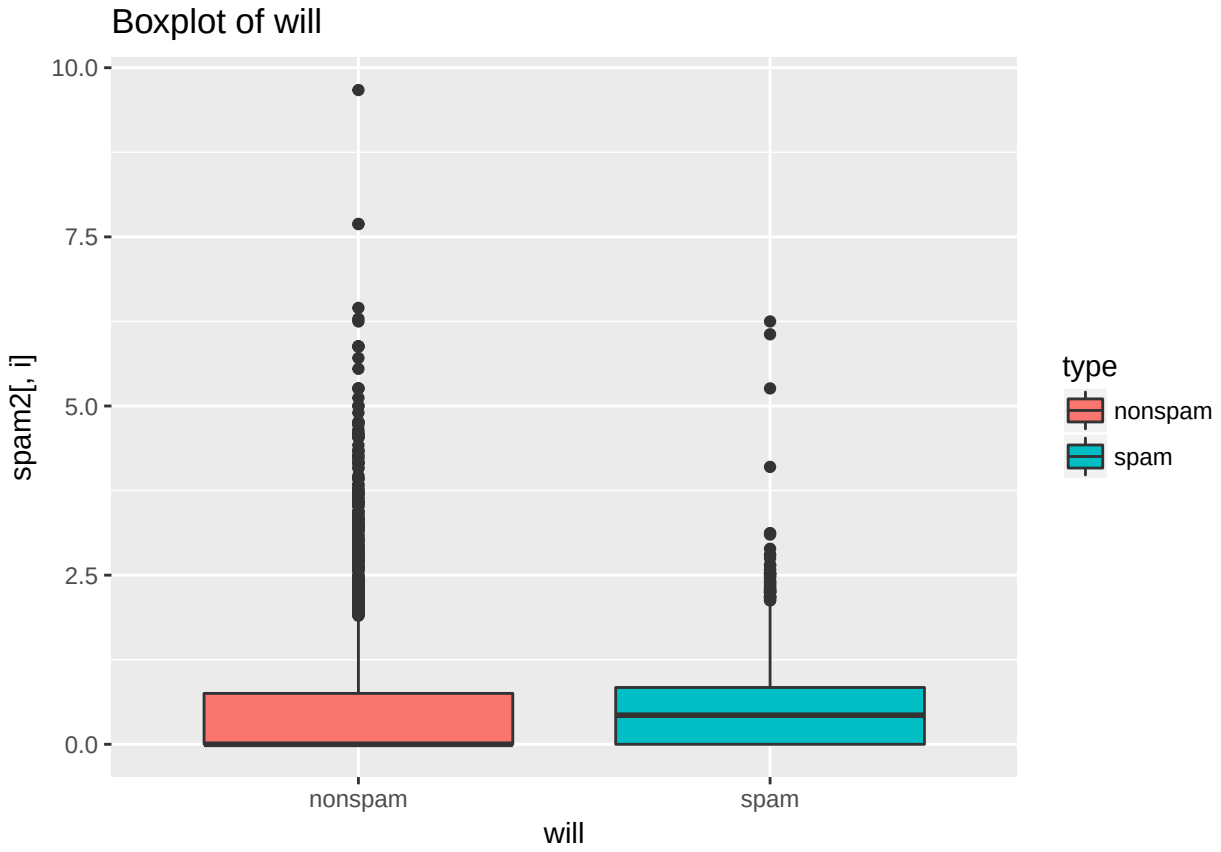
```
## Box Plots
spam2 = spam %>% select(your, you, all, our, will, type)
for(i in 1:5){
  g2 = ggplot(spam2, mapping = aes(x=type, y=spam2[,i], fill=type))
  g2_plot = g2 + geom_boxplot() +
    xlab(names(spam2)[i]) + ggtitle(paste0("Boxplot of ", names(spam2)[i]))
  print(g2_plot)
}
```


Boxplot of your
spam2[, i]
9
6
3
0
nonspam
spam
your
type
nonspam
spam


Boxplot of you
spam2[, i]
15
10
5
0
nonspam
spam
you
type
nonspam
spam


Boxplot of all
spam2[, i]
5
4
3
2
1
0
nonspam
spam
all
type
nonspam
spam


Boxplot of our
spam2[, i]
10.0
7.5
5.0
2.5
0.0
nonspam
spam
our
type
nonspam
spam

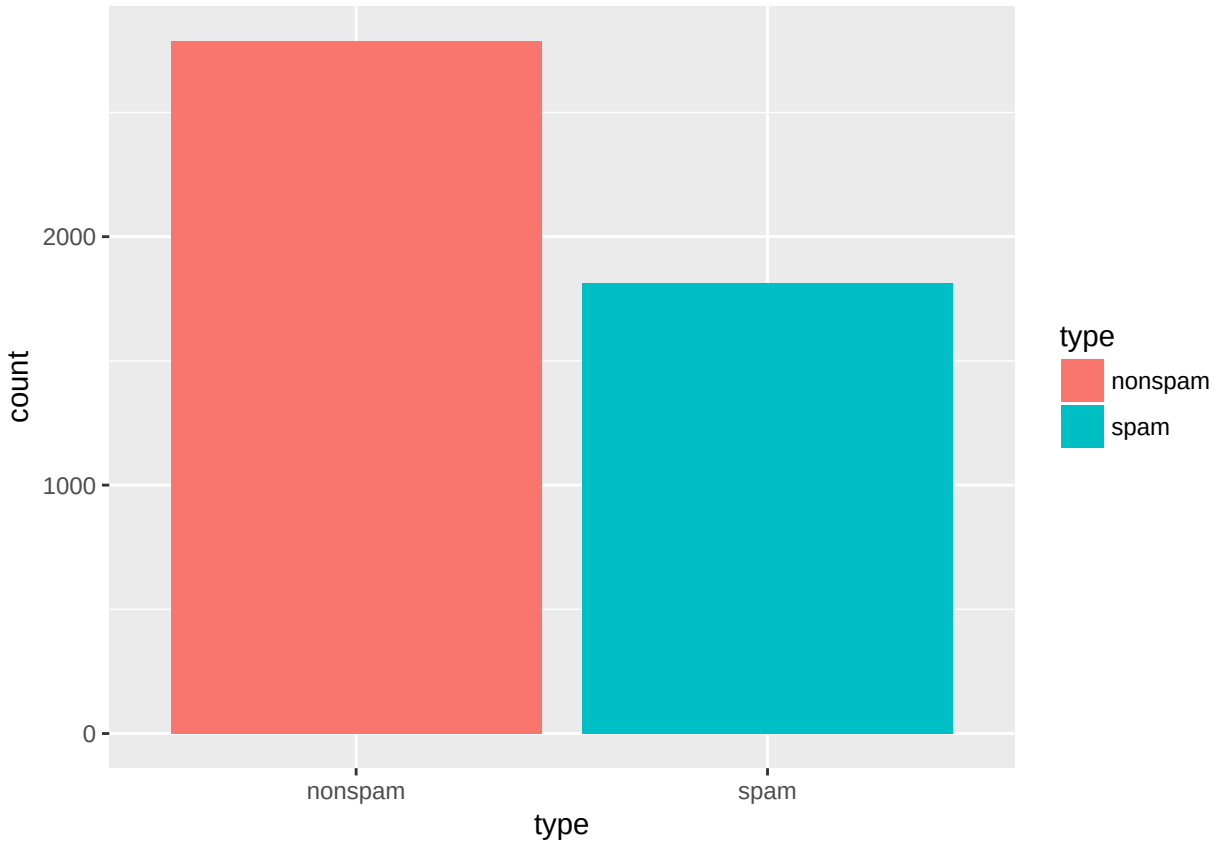



```
names(spam2)[1]
```

```
## [1] "your"
```

Bar Plot of Type

```
g = ggplot(spam, mapping = aes(x=type, fill=type))
g + geom_bar()
```



Trees

Error in tree(type ~ ., split = "gini", data = spam.train) : maximum depth reached

Doing a normal tree doesn't make sense – there are too many terminal nodes... R won't let you proceed as the maximum depth is reached.

Random Forest

```
###################################
#####Bagging & Random Forests#####
###################################
library(randomForest)


#Splitting the data into training and test sets by an 70% - 30% split.
set.seed(0)
train = sample(1:nrow(spam), 7*nrow(spam)/10) #Training indices.
spam.test = spam[-train, ] #Test dataset.
type.test = type[-train] #Test response.

#Fitting an initial random forest to the training subset.
set.seed(0)
email = randomForest(type ~ ., data = spam, subset = train, importance = TRUE)
email
```

```r
#The MSE and percent variance explained are based on out-of-bag estimates,
#yielding unbiased error estimates. The model reports that mtry = 4, which is
#the number of variables randomly chosen at each split. Since we have 13 overall
#variables, we could try all 13 possible values of mtry. We will do so, record
#the results, and make a plot.

#Varying the number of variables used at each step of the random forest procedure.
set.seed(0)
oob.err = numeric(57)
for (mtry in 1:57) {
  fit = randomForest(type ~ ., data = spam[train, ], mtry = mtry)
  oob.err[mtry] = fit$err.rate[500]
  cat("We're performing iteration", mtry, "\n")
}

#Visualizing the OOB error.
plot(1:57, oob.err, pch = 16, type = "b",
     xlab = "Variables Considered at Each Split",
     ylab = "OOB Mean Squared Error",
     main = "Random Forest OOB Error Rates\nby # of Variables")

#Can visualize a variable importance plot.
importance(email)
varImpPlot(email)

min.mtry = 8
set.seed(0)
email.total = randomForest(type ~ ., data = spam, mtry = min.mtry, subset = train, importance = TRUE)


#Ftting and visualizing a classification tree to the training data.
plot(email.total)
text(email.total, pretty = 0)
summary(email.total)
email.total

importance(email.total)
varImpPlot(email.total)
#Using the trained decision tree to classify the test data.
tree.pred = predict(email.total, spam.test, type = "class")
tree.pred

#Assessing the accuracy of the overall tree by constructing a confusion matrix.
table(tree.pred, spam.test$type)
(50 + 27)/(807 + 497 + 50 + 27)
```

Performance of Random Forest:

email.total

Call: randomForest(formula = type ~ ., data = spam, mtry = min.mtry, importance = TRUE, subset = train) Type of random forest: classification Number of trees: 500 No. of variables tried at each split: 8

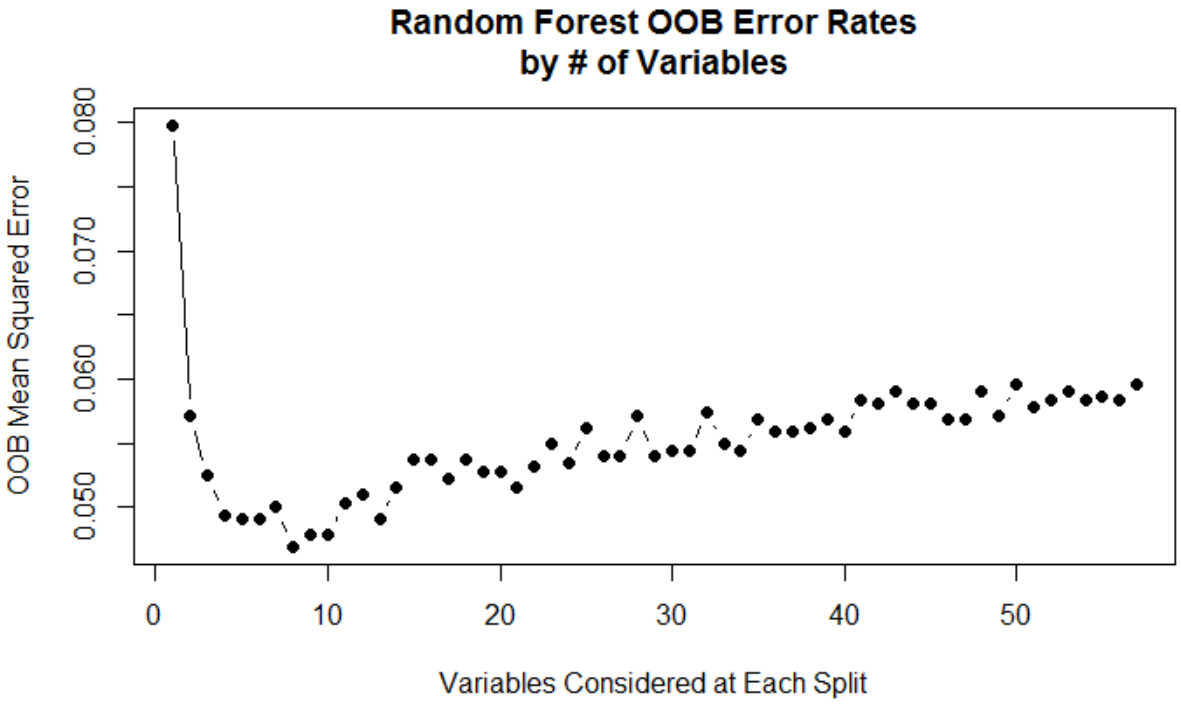


Figure 1:

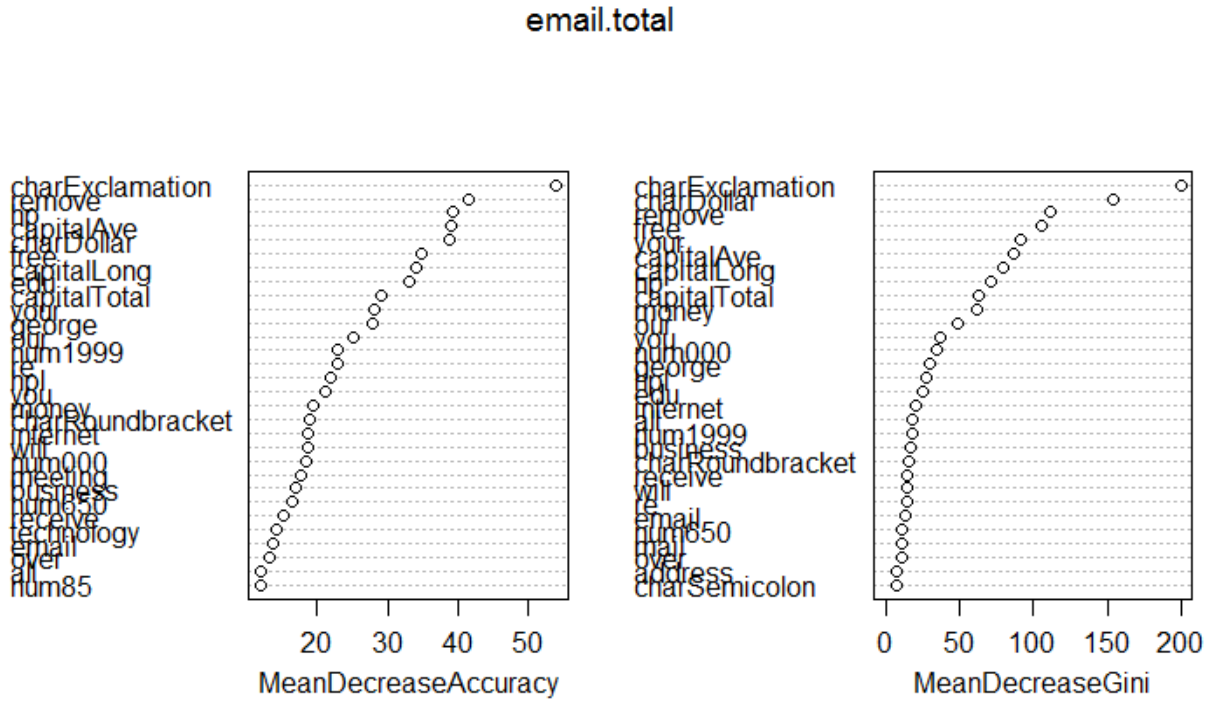


Figure 2:

```
    OOB estimate of  error rate: 4.91%
```

Confusion matrix: nonspam spam class.error nonspam 1895 59 0.03019447 spam 99 1167 0.07819905

confusionMatrix(tree.pred, spam.test$type) Confusion Matrix and Statistics

```
      Reference
```

Prediction nonspam spam nonspam 807 50 spam 27 497

```
           Accuracy : 0.9442
             95% CI : (0.9308, 0.9558)
No Information Rate : 0.6039
P-Value [Acc > NIR] : < 2e-16

              Kappa : 0.8826
```

Mcnemar's Test P-Value : 0.01217

```
        Sensitivity : 0.9676
        Specificity : 0.9086
     Pos Pred Value : 0.9417
     Neg Pred Value : 0.9485
         Prevalence : 0.6039
     Detection Rate : 0.5844
```

Detection Prevalence : 0.6206
Balanced Accuracy : 0.9381

```
   'Positive' Class : nonspam
```

GBM

```
library(kernlab)
library(gbm)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.1
```

```
data(spam)
spam_scaled <- scale(spam[,-58])
spam_scaled <- as.data.frame(spam_scaled)
train <- sample(c(1:nrow(spam_scaled)), 0.7*nrow(spam_scaled))
spam_scaled$type <- spam$type
spam_scaled$type <- as.numeric(spam_scaled$type)-1  #1: spam; 0: nonspam
sp_train <- spam_scaled[train,]
sp_test <- spam_scaled[-train,]
```

```
set.seed(0)
boost_sp <- gbm(type ~ ., data = sp_train,
                distribution = "bernoulli",
                n.trees = 10000,
                interaction.depth = 4,
                shrinkage = 0.001)
```

```
set.seed(0)
boost_sp <-readRDS("tree.RDS")
n_trees <- seq(from = 100, to = 10000, by = 100)
predmat <- predict(boost_sp, newdata = sp_test, n.trees = n_trees, type = "response")
predmat <- round(predmat)
berr <- with(sp_test, apply((predmat - type)^2, 2, mean))
which.min(berr)
```

```
## 9400
##   94
```

```
1-min(berr)
```

```
## [1] 0.9522085
```

```
pred <- predmat[,which.min(berr)]
pred <- as.factor(pred)
truth <- sp_test$type
table(truth, pred)
```

```
##      pred
## truth   0   1
##     0 806  27
##     1  39 509
```

```
accuracy = (table(truth, pred)[1,1]+table(truth, pred)[2,2])/1381
accuracy
```

```
## [1] 0.9522085
```

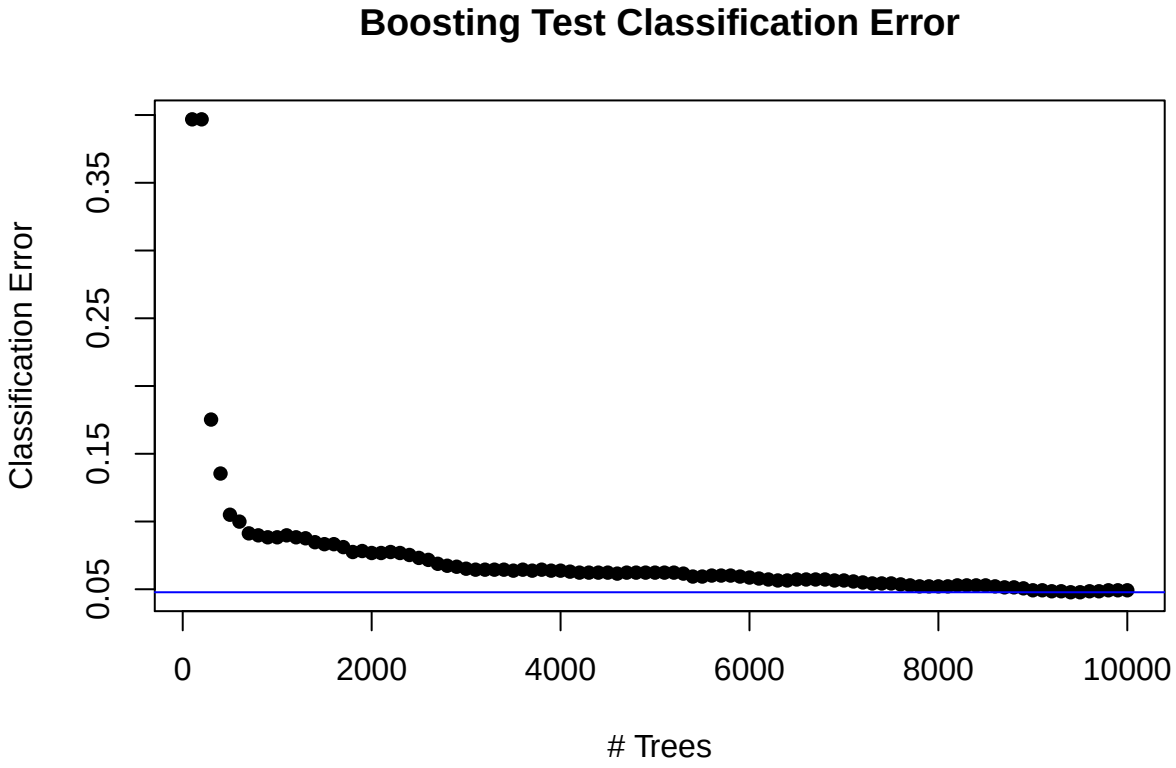
```
sensitivity = (table(truth, pred)[2,2])/(table(truth, pred)[2,2]+table(truth, pred)[2,1]) #tpr
sensitivity
```

```
## [1] 0.9288321
```

```
specificity = (table(truth, pred)[1,1])/(table(truth, pred)[1,1]+table(truth, pred)[1,2]) #tnr
specificity
```

```
## [1] 0.967587
```

```
plot(n_trees, berr, pch = 16,
     ylab = "Classification Error",
     xlab = "# Trees",
     main = "Boosting Test Classification Error")
abline(h = min(berr), col = "blue")  #boosted
```


Boosting Test Classification Error
Classification Error
0.05
0.15
0.25
0.35
0
2000
4000
6000
8000
10000
Trees