

# spamData.R

Admin

2021-01-20

```
library(kernlab)
```

```
## Warning: package 'kernlab' was built under R version 4.0.3
```

```
data(spam)
str(spam[,1:5])
```

```
## 'data.frame': 4601 obs. of 5 variables:
## $ make : num 0 0.21 0.06 0 0 0 0 0 0.15 0.06 ...
## $ address: num 0.64 0.28 0 0 0 0 0 0 0.12 ...
## $ all : num 0.64 0.5 0.71 0 0 0 0 0 0.46 0.77 ...
## $ num3d : num 0 0 0 0 0 0 0 0 0 ...
## $ our : num 0.32 0.14 1.23 0.63 0.63 1.85 1.92 1.88 0.61 0.19 ...
```

```
#generando subset de prueba
set.seed(3435)
```

```
# Distribución de variables en valores booleanos como trainIndicator
trainIndicator <- rbinom(4601, size = 1, prob = 0.5)
table(trainIndicator)
```

```
## trainIndicator
## 0 1
## 2314 2287
```

```
#Se separan el dataset en Test y Training dataset mediante
# distribución probabilística rbinom
trainSpam = spam[trainIndicator == 1,]
testSpam = spam[trainIndicator == 0,]

names(trainSpam)
```

```
## [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"
```

```
head(trainSpam)
```

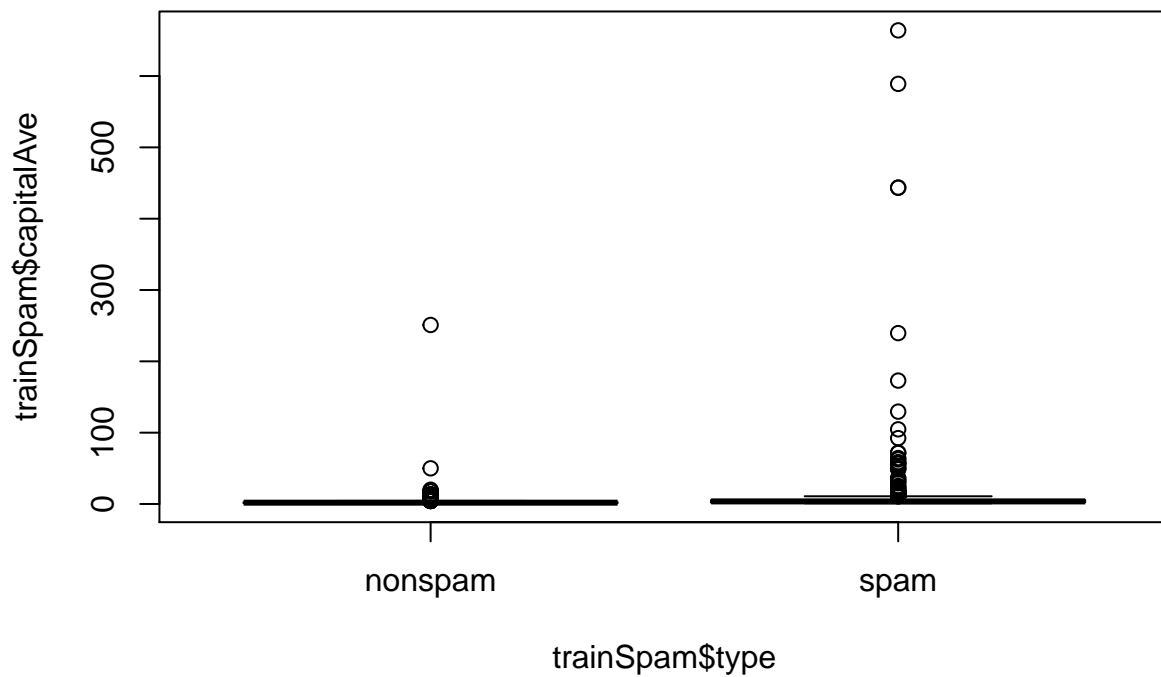
```
##      make address  all num3d  our over remove internet order mail receive will
## 1  0.00    0.64 0.64    0 0.32 0.00    0.00          0 0.00 0.00    0.00 0.64
## 7  0.00    0.00 0.00    0 1.92 0.00    0.00          0 0.00 0.64    0.96 1.28
## 9  0.15    0.00 0.46    0 0.61 0.00    0.30          0 0.92 0.76    0.76 0.92
## 12 0.00    0.00 0.25    0 0.38 0.25    0.25          0 0.00 0.00    0.12 0.12
## 14 0.00    0.00 0.00    0 0.90 0.00    0.90          0 0.00 0.90    0.90 0.00
## 16 0.00    0.42 0.42    0 1.27 0.00    0.42          0 0.00 1.27    0.00 0.00
##      people report addresses free business email  you credit your font num000
## 1  0.00    0          0 0.32          0 1.29 1.93    0.00 0.96    0      0
## 7  0.00    0          0 0.96          0 0.32 3.85    0.00 0.64    0      0
## 9  0.00    0          0 0.00          0 0.15 1.23    3.53 2.00    0      0
## 12 0.12    0          0 0.00          0 0.00 1.16    0.00 0.77    0      0
## 14 0.90    0          0 0.00          0 0.00 2.72    0.00 0.90    0      0
## 16 0.00    0          0 1.27          0 0.00 1.70    0.42 1.27    0      0
##      money hp hpl george num650 lab labs telnet num857 data num415 num85
## 1  0.00 0 0      0      0 0 0      0      0 0.00    0      0
## 7  0.00 0 0      0      0 0 0      0      0 0.00    0      0
## 9  0.15 0 0      0      0 0 0      0      0 0.15    0      0
## 12 0.00 0 0      0      0 0 0      0      0 0.00    0      0
## 14 0.00 0 0      0      0 0 0      0      0 0.00    0      0
## 16 0.42 0 0      0      0 0 0      0      0 0.00    0      0
##      technology num1999 parts pm direct cs meeting original project re edu table
## 1  0      0 0.00    0 0 0.00 0      0      0.0    0 0 0      0
## 7  0      0 0.00    0 0 0.00 0      0      0.0    0 0 0      0
## 9  0      0 0.00    0 0 0.00 0      0      0.3    0 0 0      0
## 12 0      0 0.00    0 0 0.00 0      0      0.0    0 0 0      0
## 14 0      0 0.00    0 0 0.00 0      0      0.0    0 0 0      0
## 16 0      1.27    0 0 0.42 0      0      0.0    0 0 0      0
##      conference charSemicolon charRoundbracket charSquarebracket charExclamation
## 1  0      0.000          0.000          0      0.778
## 7  0      0.000          0.054          0      0.164
## 9  0      0.000          0.271          0      0.181
## 12 0      0.022          0.044          0      0.663
## 14 0      0.000          0.000          0      0.000
## 16 0      0.000          0.063          0      0.572
##      charDollar charHash capitalAve capitalLong capitalTotal type
## 1  0.000    0.000      3.756          61      278 spam
## 7  0.054    0.000      1.671          4      112 spam
```

```
## 9      0.203  0.022   9.744    445    1257 spam
## 12     0.000  0.000   1.243     11     184 spam
## 14     0.000  0.000   2.083      7      25 spam
## 16     0.063  0.000   5.659     55     249 spam
```

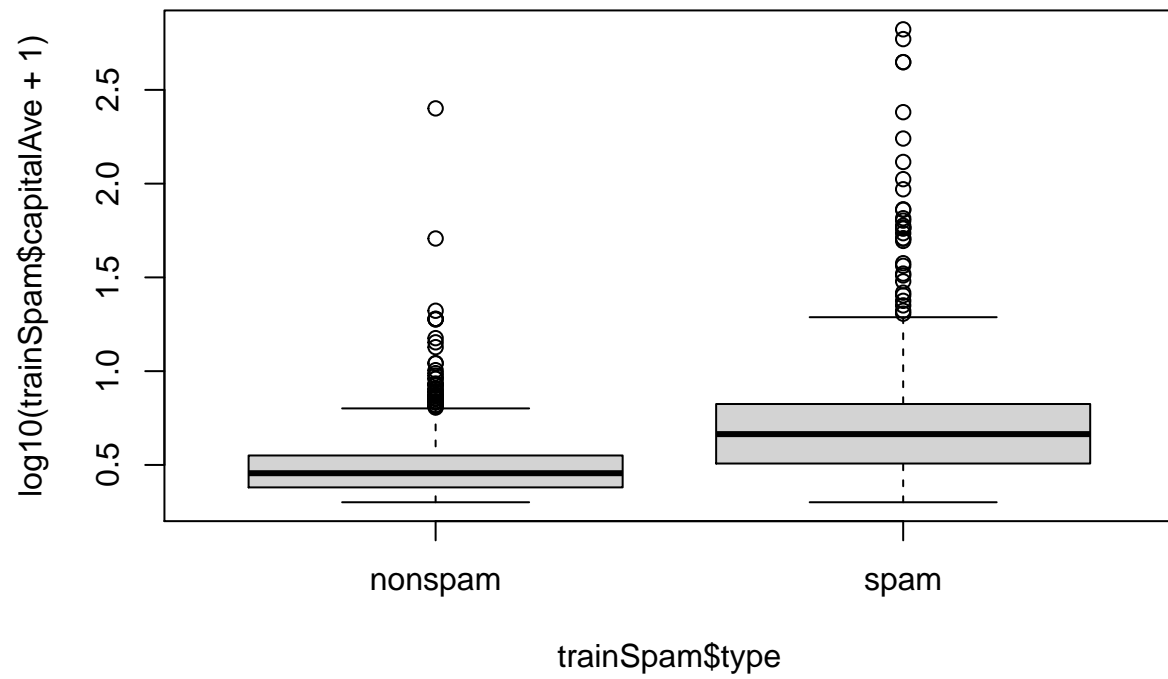
```
table(trainSpam$type)
```

```
##
## nonspam  spam
##    1381    906
```

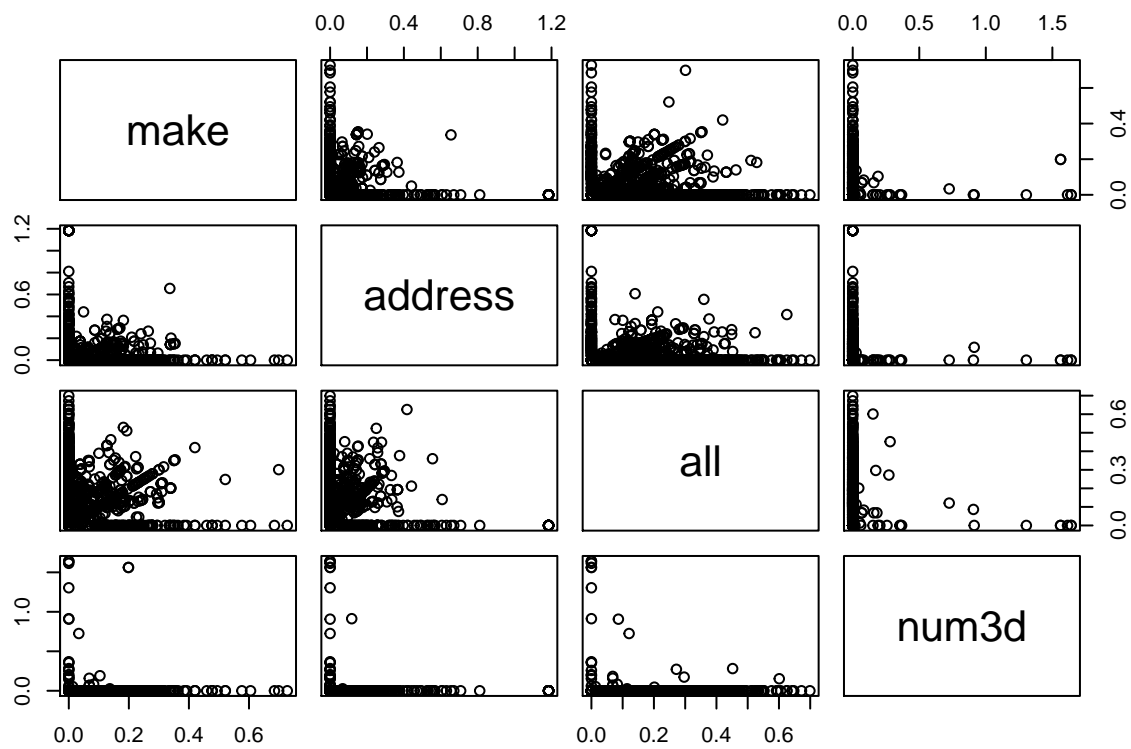
```
#Se grafica incidencia entre correos spam que contienen mayor promedio de
# letras mayúsculas en su contenido
plot(trainSpam$capitalAve ~ trainSpam$type)
```



```
#en logaritmo base 10 para mejor visualización.
plot(log10(trainSpam$capitalAve+1) ~ trainSpam$type)
```



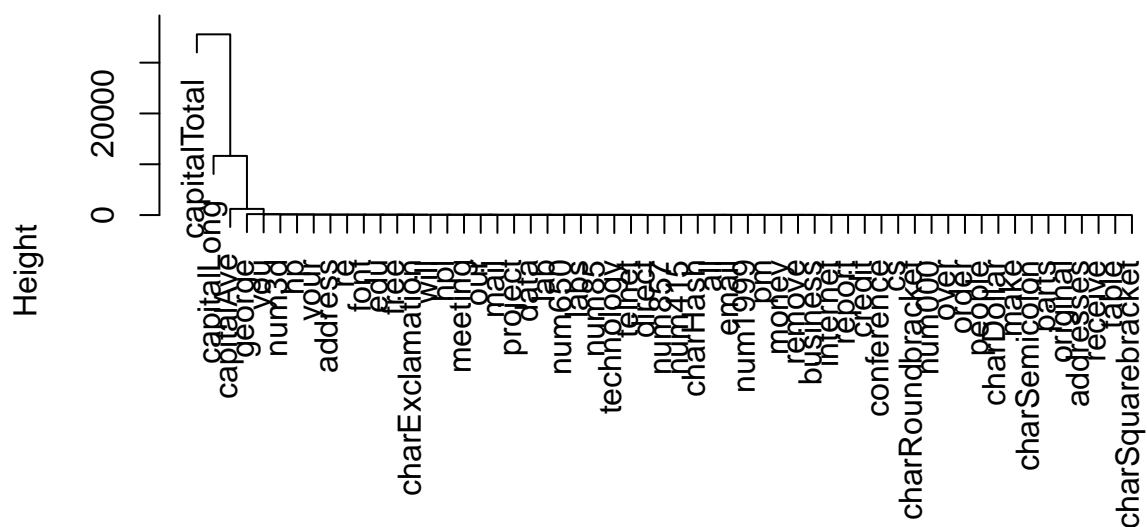
```
#Se eliminan los valores en cero para visualización  
plot(log10(trainSpam[,1:4] + 1))
```



```
#Cluster que identifica las variables con mayor incidencia en agrupación
hCluster <- hclust(dist(t(trainSpam[,1:57])))

#Gráfico de dendograma de cluster
plot(hCluster)
```

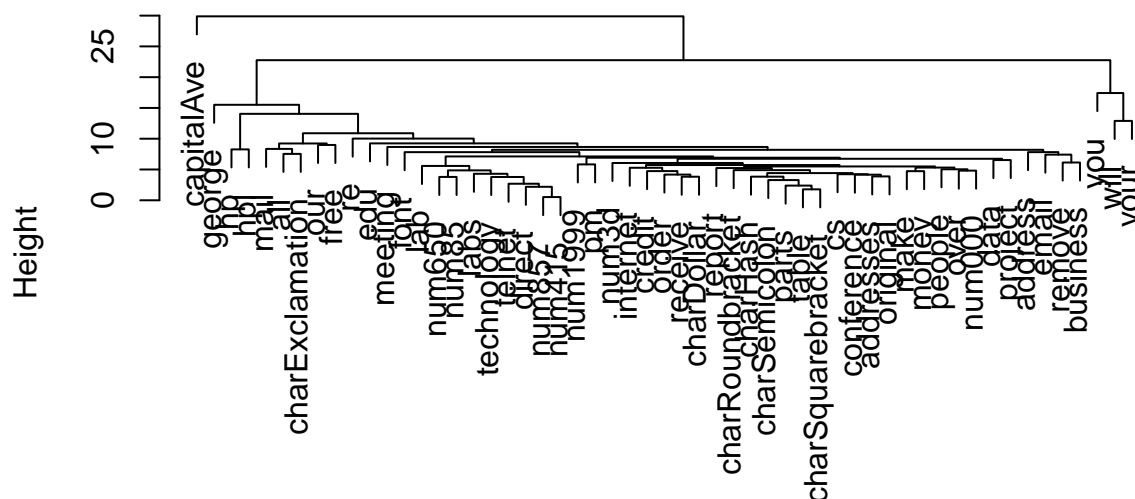
## Cluster Dendrogram



```
dist(t(trainSpam[, 1:57]))
hclust (*, "complete")
```

```
# Log Base 10
hClusterUpdated <- hclust(dist(t(log10(trainSpam[,1:55] + 1))))
#dendograma
plot(hClusterUpdated)
```

## Cluster Dendrogram



```
dist(t(log10(trainSpam[, 1:55] + 1)))
hclust (*, "complete")
```

##### STATISTICAL PREDICTION MODELLING #####

```
trainSpam$numtype = as.numeric(trainSpam$type) -1
costFunction = function(x,y) sum(x != (y > 0.5))
cvError = rep(NA,55)
library(boot)
for (i in 1:55){
  lmFormula = reformulate(names(trainSpam)[i], response = "numtype")
  glmFit = glm(lmFormula, family = "binomial", data = trainSpam)
  cvError[i] = cv.glm(trainSpam, glmFit, costFunction, 2)$delta[2]
}
```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred







```
# Cual predictor tiene un menor error de validación cruzada?  
names(trainSpam)[which.min(cvError)]
```

```
## [1] "charDollar"
```

```
#Modelo de regresión logística
```

```
predictionModel = glm(numtype ~ charDollar, family = "binomial", data = trainSpam)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## hacer predicciones sobre el set de prueba
```

```
predictionTest = predict(predictionModel, testSpam)
```

```
predictedSpam = rep("nonspam", dim(testSpam)[1])
```

```
#Clasificar como spam aquellos con una probabilidad mayor a 0.5
```

```
predictedSpam[predictionModel$fitted > 0.5] = "spam"
```

```
#Obtener una medida de incertidumbre
```

```
table(predictedSpam, testSpam$type)
```

```
##
```

```
## predictedSpam nonspam spam
```

```
##      nonspam    1346  458
```

```
##      spam        61  449
```

```
#tasa de error
```

```
(61 + 458)/(1346 + 458 + 61 + 449)
```

```
## [1] 0.2242869
```