

SonoBat Analysis v3 2021

Howard Nguyen - Salash Brahimi - Pankaj Gupta

2023-04-08

load libraries

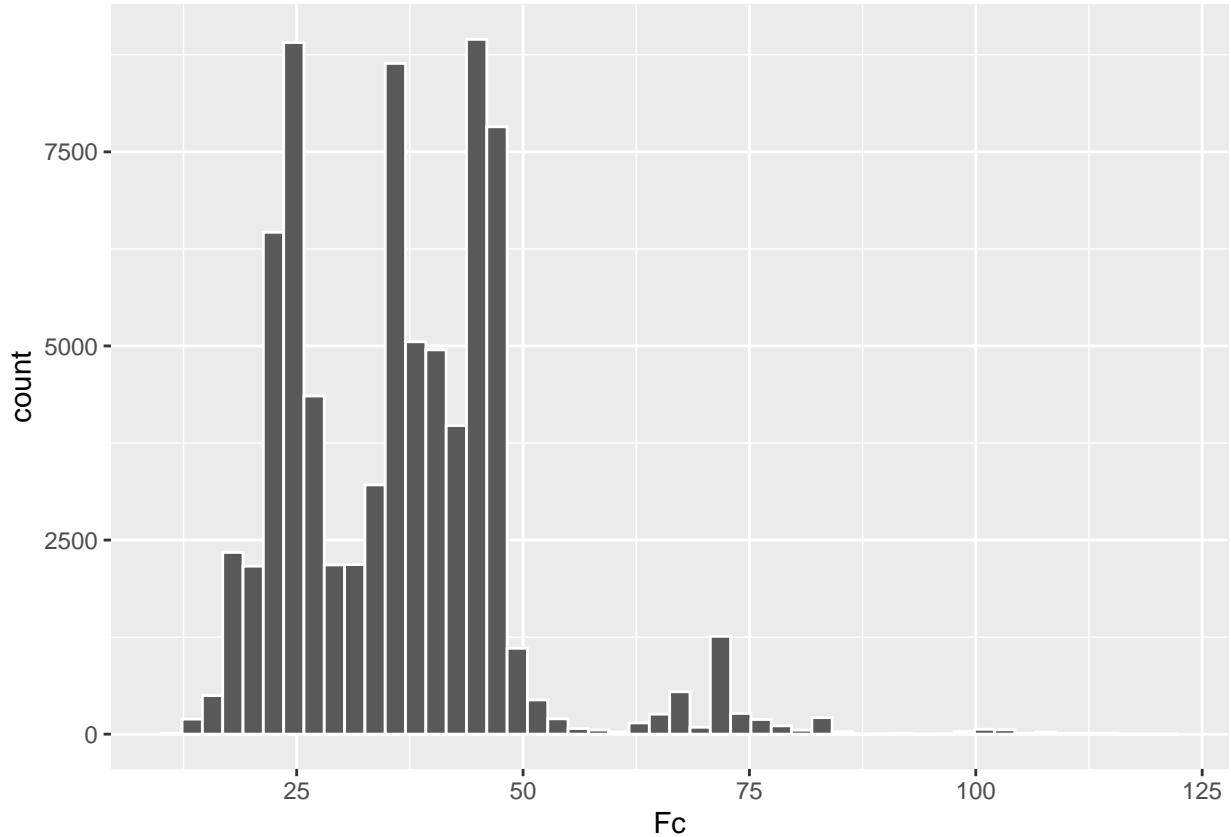
```
library(tidyverse)
library(readxl)
library(ggplot2)
library(dplyr)
library(cluster)
library(factoextra)
library(FactoMineR)
library(datasets)
library(corrplot)
# Modeling packages
library(tidymodels)
library(earth) # for fitting MARS models
library(caret) # for automating the tuning process
library(randomForest)
```

load the datasets

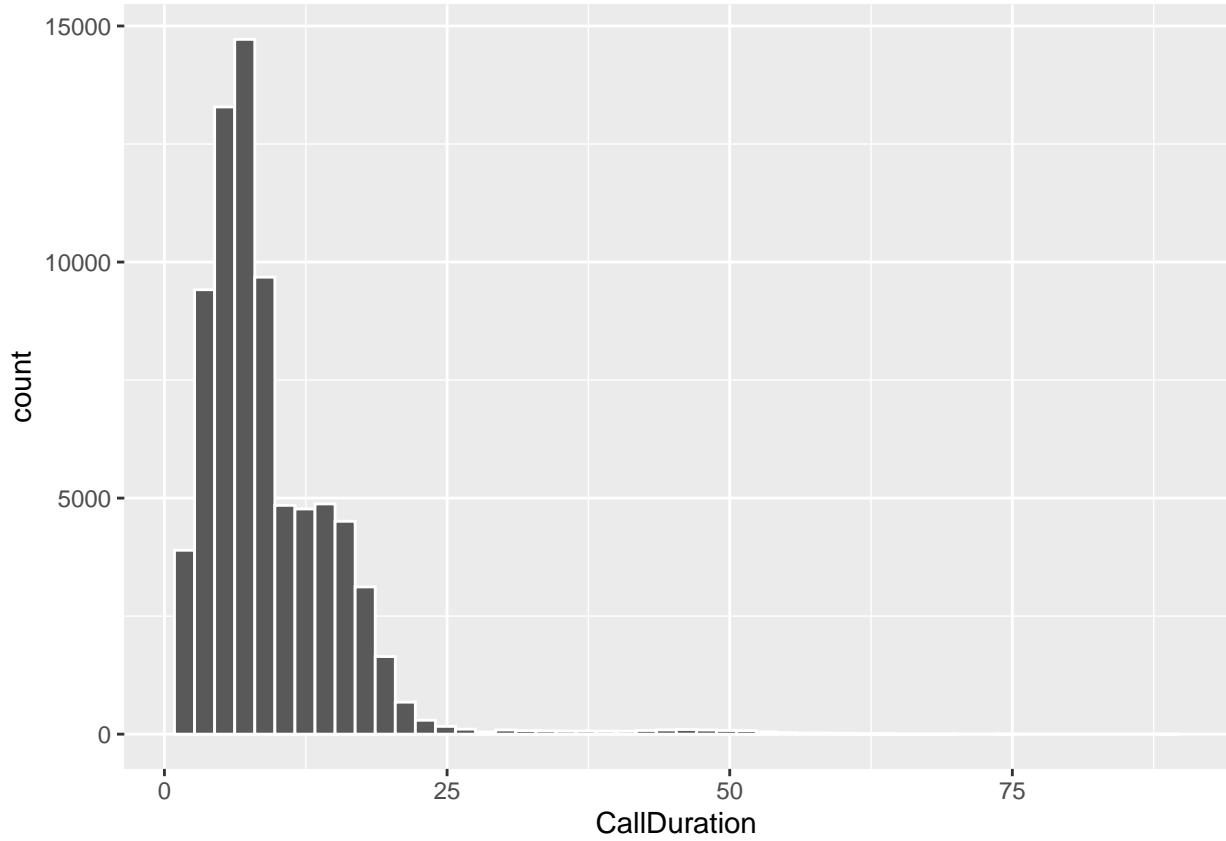
```
df22 <- read.delim("SierraLeone2022113.txt")
```

EDA

```
# exploring features of SonoBat dataset
tidymodels_prefer()
ggplot(df22, aes(x = Fc)) +
  geom_histogram(bins = 50, col = "white")
```



```
ggplot(df22, aes(x = CallDuration)) +  
  geom_histogram(bins = 50, col = "white")
```



This plot shows us that the data are right-skewed, there are much smaller numbers than the large count in Fc.

```
head(df22)
```

```
##                                     Path
## 1 E:\\SierraLeone\\AcousticData\\5-2022\\KBH-1_20220507_191026.wav
## 2 E:\\SierraLeone\\AcousticData\\5-2022\\KBH-1_20220507_191026.wav
## 3 E:\\SierraLeone\\AcousticData\\5-2022\\KBH-1_20220507_191026.wav
## 4 E:\\SierraLeone\\AcousticData\\5-2022\\KBH-1_20220507_191026.wav
## 5 E:\\SierraLeone\\AcousticData\\5-2022\\KBH-1_20220507_191026.wav
## 6 E:\\SierraLeone\\AcousticData\\5-2022\\KBH-1_20220507_191026.wav
##           Filename TimeInFile PrecedingIntrvl CallsPerSec CallDuration
## 1 KBH-1_20220507_191026.wav      1339          174  5.571619   13.73850
## 2 KBH-1_20220507_191026.wav      691          172  5.571619   14.95602
## 3 KBH-1_20220507_191026.wav     1514          176  5.571619   14.27390
## 4 KBH-1_20220507_191026.wav     2346          171  5.571619   11.26184
## 5 KBH-1_20220507_191026.wav     1645          130  5.571619   13.28864
## 6 KBH-1_20220507_191026.wav     2519          175  5.571619   14.87257
##           Fc    HiFreq  LowFreq Bndwdth FreqMaxPwr PrcntMaxAmpDur TimeFromMaxToFc
## 1 30.09062 31.75588 29.66113 2.094757 30.69036      52.51447  5.839990
## 2 30.47646 32.04760 30.19313 1.854476 31.32279      38.19162  8.302497
## 3 30.12007 32.11109 29.89022 2.220872 30.23859      80.33998  2.617597
## 4 30.41762 31.66190 30.17061 1.491294 30.41394      92.87591 -0.047009
## 5 30.89197 32.88805 30.74737 2.140684 31.07854      74.30500  2.896576
## 6 29.93181 31.70977 29.91662 1.793147 30.28021      75.74210  2.900367
```

```

##   FreqKnee PrcntKneeDur   StartF      EndF DominantSlope SlopeAtFc StartSlope
## 1 30.40580     69.79122 31.75588 29.76242      0.102458  0.374462 -0.397281
## 2 30.39610     91.02874 31.81982 30.55663      0.125534  0.087964 -0.110441
## 3 30.59488     58.76594 32.11109 30.24524      0.105369  0.070998 -0.848749
## 4 30.50557     67.62201 31.65980 30.17061      0.056590  0.054453 -0.083356
## 5 32.23373     12.73800 32.88805 30.90555      0.148622  0.093604 -0.147918
## 6 30.55992     45.68915 31.55084 30.47441      0.102887  0.197388 -0.120748
##   EndSlope SteepestSlope LowestSlope TotalSlope HiFtoKnSlope KneeToFcSlope
## 1 0.079700     0.402521  0.012330 0.121189    0.120307  0.160323
## 2 -0.240827     0.299449  0.002555 0.127720    0.131206  0.204417
## 3 0.274263     0.474276  0.003707 0.129362    0.129874  0.105438
## 4 -0.336729     0.370681  0.000328 0.121868    0.126511  0.060584
## 5 0.470184     0.502855  0.002062 0.148062    0.451635  0.130891
## 6 0.384728     0.398594  0.001158 0.119814    0.151485  0.081653
##   CummNmLzdSlp HiFtoFcExpAmp HiFtoFcDmp KnToFcExpAmp KnToFcDmp HiFtoKnExpAmp
## 1 0.132652     0.238721  0.144117 0.066988  0.463974  0.195406
## 2 0.143118     0.212305  0.149537 0.021113  1.278390  0.260004
## 3 0.136665     0.233137  0.143647 0.029836  0.495926  0.185551
## 4 0.114130     0.184034  0.184496 0.041912  0.427134  0.253072
## 5 0.154234     0.171928  0.192721 0.168279  0.196062  0.157423
## 6 0.132191     0.209137  0.157434 0.161267  0.186608  0.204758
##   HiFtoKnDmp FreqLedge LedgeDuration LdgToFcSlp KnToFcDur UpprKnFreq
## 1 0.193069     29.81975   0.919608 0.425551  3.466423  31.47637
## 2 0.142817     30.39610   0.400169 0.204417  0.400169  31.90886
## 3 0.233254     30.00565   0.471639 0.233301  5.697055  31.52909
## 4 0.200184     30.38093   0.540605 0.091026  2.797044  31.33969
## 5 0.951423     30.85924   0.164799 0.248427  11.077992 32.38654
## 6 0.273281     30.03805   0.565925 0.238914  7.370012  31.45446
##   HiFtoUpprKnSlp HiFtoUpprKnAmp HiFtoUpprKnExp HiFtoKnAmp HiFtoKnExp HiFtoFcAmp
## 1 0.311719     32.55108   -0.009859 31.82506 -0.003891  31.83766
## 2 0.223328     32.66302   -0.006978 32.32240 -0.004222  32.29787
## 3 0.500066     33.33506   -0.015739 32.03484 -0.004175  32.03505
## 4 0.357447     33.58187   -0.011336 32.20854 -0.004072  32.17363
## 5 0.446222     34.23316   -0.013637 34.25621 -0.013858  32.97719
## 6 0.339414     32.87662   -0.010728 32.06753 -0.004877  31.85910
##   HiFtoFcExp UpprKnToKnAmp UpprKnToKnExp KnToFcAmp KnToFcExp LdgToFcAmp
## 1 -0.003944     32.47378   -0.005367 32.45210 -0.005322  24.13833
## 2 -0.004114     28.56538   0.004035 27.35948  0.006704  27.35948
## 3 -0.004206     31.59220   -0.003193 31.71317 -0.003492  26.53478
## 4 -0.003946     31.42564   -0.002259 31.30889 -0.001991  29.06128
## 5 -0.004699     32.93767   -0.004843 32.77774 -0.004168  27.26371
## 6 -0.003919     31.17619   -0.002245 31.33902 -0.002699  34.21957
##   LdgToFcExp FreqCtr FBak32dB FFwd32dB FBak20dB FFwd20dB FBak15dB FFwd15dB
## 1 0.014223     30.71671 29.68433 32.01823 30.04900 31.58063 30.19487 31.43476
## 2 0.006704     30.94288 29.87336 32.28467 30.27525 31.94976 30.47619 31.81580
## 3 0.007740     30.77138 29.68421 32.42105 30.03509 31.64912 30.17544 31.50877
## 4 0.002993     30.89845 15.23173 34.38274 30.10717 31.88866 30.28532 31.71051
## 5 0.008046     31.45177 30.53004 32.94228 30.83157 32.48999 30.98233 32.33922
## 6 -0.007970    30.53643 29.52291 32.01685 29.92733 31.54502 30.06214 31.41022
##   FBak5dB FFwd5dB Bndw32dB Bndw20dB Bndw15dB Bndw5dB Dur0f32dB Dur0f20dB
## 1 30.48661    31.14302 2.333903 1.531624 1.239886 0.656410 0.282956 11.287954
## 2 30.81109    31.54788 2.411303 1.674516 1.339613 0.736787 0.023539 12.413966
## 3 30.45614    31.22807 2.736842 1.614035 1.333333 0.771930 0.023582 11.399816
## 4 30.64161    31.35421 19.151009 1.781489 1.425191 0.712596 0.026143 0.026143

```

```

## 5 31.28386 32.03769 2.412250 1.658422 1.356890 0.753828 0.023543 10.629694
## 6 30.33175 31.07320 2.493944 1.617694 1.348078 0.741443 0.023580 11.598096
## Dur0f15dB Dur0f5dB Amp1stQrtl Amp2ndQrtl Amp3rdQrtl Amp4thQrtl Amp1stMean
## 1 10.039495 6.086671 130.28833 17281.554 16891.655 12801.14 0.929226
## 2 10.391252 5.523897 124.37214 20350.936 18764.732 14174.52 0.903128
## 3 10.009441 6.015063 106.51411 13988.481 19669.105 19252.03 0.829289
## 4 0.379221 5.993666 86.20392 5960.607 7991.483 16983.38 0.762675
## 5 9.405933 5.195669 124.12725 17704.742 18487.054 17101.47 0.896930
## 6 11.032982 5.008191 132.08999 16954.333 18765.954 16800.62 0.903175
## Amp2ndMean Amp3rdMean Amp4thMean LnExpA_StartAmp LnExpB_StartAmp
## 1 0.968364 0.964800 0.899538 -0.005587 -0.290485
## 2 0.973312 0.972437 0.903134 0.011588 -0.223018
## 3 0.898427 0.960601 0.952779 0.016248 -0.760481
## 4 0.764390 0.813589 0.921102 0.013607 -1.331607
## 5 0.946568 0.957132 0.949112 0.008725 -0.468741
## 6 0.924924 0.936323 0.897257 0.033862 -1.229953
## AmpStartLn60ExpC LnExpA_EndAmp LnExpB_EndAmp AmpEndLn60ExpC AmpK.start
## 1 4.094345 0.001690 -0.200338 4.094345 5.595655
## 2 4.094345 0.019821 -0.511712 4.094345 3.833285
## 3 4.094345 0.005645 -0.276453 4.094345 21.828047
## 4 4.094345 -0.111835 -0.234212 4.094345 77.365796
## 5 4.094345 -0.015154 -0.148069 4.094345 12.172568
## 6 4.094345 -0.045308 -0.094687 4.094345 17.920305
## AmpK.end AmpKurtosis AmpSkew AmpVariance AmpMoment AmpGausR2 PreFc250
## 1 3.223241 13.382612 -2.801709 0.002822 0.002817 0.512548 0.268184
## 2 6.884980 8.804000 -2.316084 0.004054 0.004048 0.658756 -0.114148
## 3 3.086202 5.570938 -1.234727 0.004108 0.004101 0.813091 -0.229106
## 4 1.667381 3.249620 0.901986 0.005819 0.005806 0.542832 0.022880
## 5 2.221442 12.565960 -2.539664 0.001987 0.001984 0.528395 0.105247
## 6 1.264784 10.781834 -2.395888 0.004145 0.004139 0.291348 -0.211283
## PreFc500 PreFc1000 PreFc250Residue PreFc500Residue PreFc1000Residue
## 1 0.504507 0.363335 1.35e-04 0.0002860 0.001719
## 2 0.267900 0.293707 7.23e-05 0.0006100 0.000711
## 3 0.265423 0.297202 2.36e-05 0.0013060 0.001049
## 4 0.088360 0.002960 3.46e-06 0.0000192 0.000207
## 5 -0.013381 0.147683 6.07e-05 0.0002530 0.000945
## 6 -0.242640 -0.069328 2.31e-05 0.0000158 0.001027
## PreFc3000 PreFc3000Residue KneeToFcResidue Kn.FcCurviness meanKn.FcCurviness
## 1 -0.172370 0.010882 0.009812 -0.0030160 -2.08e-05
## 2 0.261109 0.000770 0.000621 -0.0096980 -6.47e-04
## 3 -0.071040 0.007726 0.005156 0.0054630 2.28e-05
## 4 -0.058375 0.001021 0.001075 -0.0000751 -6.42e-07
## 5 -0.117347 0.004350 0.003345 0.0071160 1.52e-05
## 6 -0.111972 0.003669 0.002871 0.0053300 1.71e-05
## Kn.FcCurvinessTrndSlp Quality HiFminusStartF FcMinusEndF RelPwr2ndTo1st
## 1 -5.910e-05 0.609682 0.000000 0.328197 0.254055
## 2 -3.044e-03 0.741729 0.227784 -0.080171 0.205209
## 3 -1.270e-05 0.713494 0.000000 -0.125175 0.112483
## 4 -4.950e-06 0.672488 0.002100 0.247011 0.149517
## 5 3.270e-06 0.644394 0.000000 -0.013580 0.220801
## 6 8.250e-06 0.692571 0.158924 -0.542607 0.288667
## RelPwr3rdTo1st ParentDir NextDirUp Filter Preemphasis MinAccpQuality
## 1 0 May-22 AcousticData 5 kHz medium 0.8
## 2 0 May-22 AcousticData 5 kHz medium 0.8

```

```

## 3          0 May-22 AcousticData 5 kHz    medium      0.8
## 4          0 May-22 AcousticData 5 kHz    medium      0.8
## 5          0 May-22 AcousticData 5 kHz    medium      0.8
## 6          0 May-22 AcousticData 5 kHz    medium      0.8
##   MaxSegLngth Max.CallsConsidered
## 1      0.5 sec            32
## 2      0.5 sec            32
## 3      0.5 sec            32
## 4      0.5 sec            32
## 5      0.5 sec            32
## 6      0.5 sec            32

# Remove NA values
df22 <- na.omit(df22)

df22 <- read.delim("SierraLeone2022113.txt") |>
  mutate(across(where(is_character), as_factor))
# print(df22)

#summary(df22)

# remove those attributes have zero variance and character cols
df22_new <- df22[, !(names(df22) %in% c("AmpStartLn60ExpC", "AmpEndLn60ExpC", "NextDirUp", "Preemphasis",
                                           "MinAccpQuality", "MaxSegLngth", "Max.CallsConsidered",
                                           "Path", "Filename", "ParentDir", "Filter"))]
dim(df22_new)

## [1] 77207 102

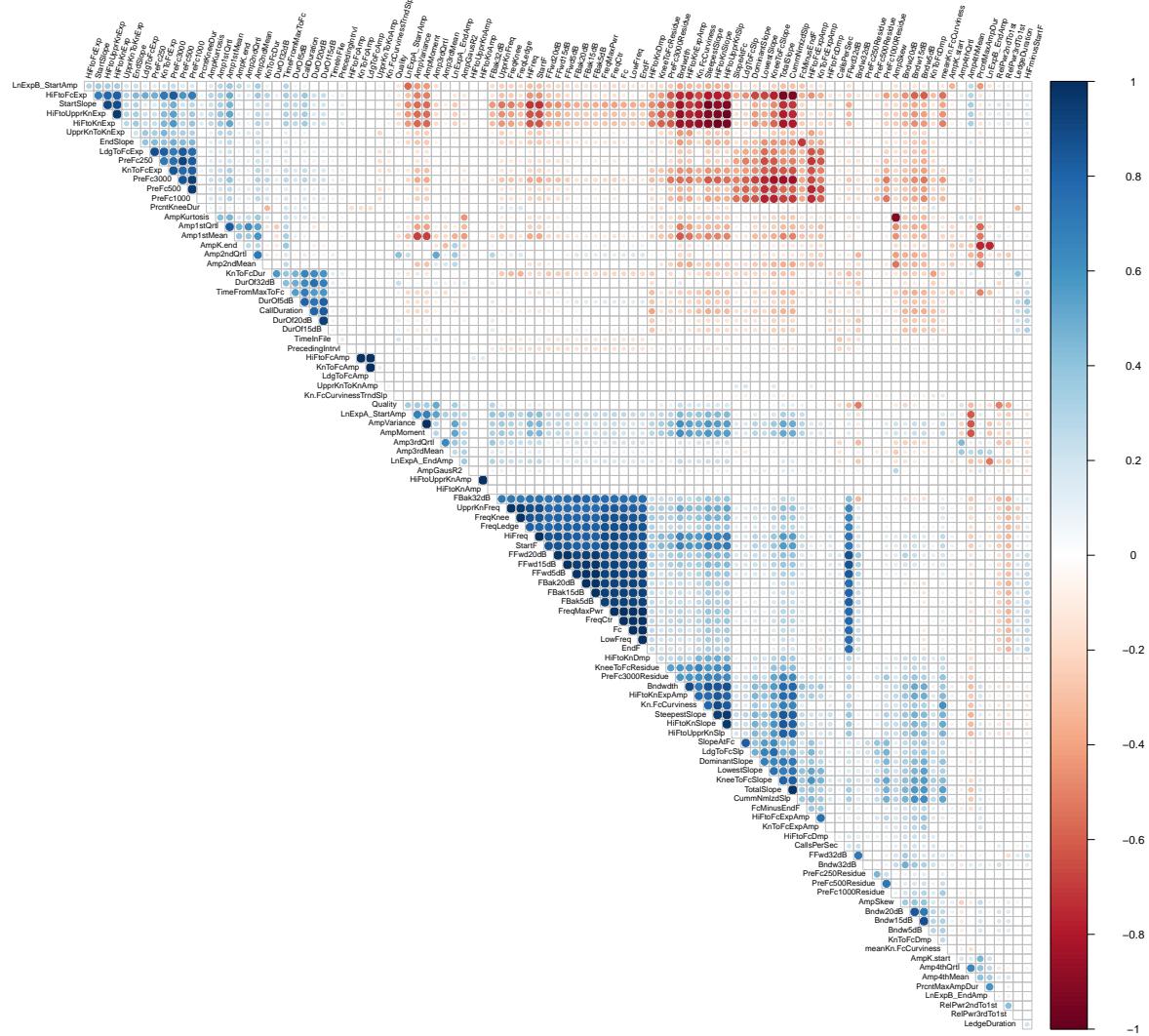
```

Visualize correlations between attributes

```

# visualize the correlation in the new dataset
corrMatrix <- stats::cor(df22_new[, 1:ncol(df22_new)])
corrplot(corrMatrix,
          order = "hclust", # order for labels, can be "original"
          type = "upper", # matrix: full, upper, lower
          diag = F, # remove diagonal
          tl.cex = 0.6, # font size
          tl.srt = 75, # label angel
          tl.col = "black", addrect = 8)

```



SPLIT THE DATA

```
# Split the data into training and testing sets
set.seed(123)
train_idx <- sample(nrow(df22_new), 0.8 * nrow(df22_new))
train_data <- df22_new[train_idx, ]
test_data <- df22_new[-train_idx, ]

# Train the random forest model
rf_model <- randomForest(Fc ~ ., data = train_data, ntree = 5)
```

```

# Evaluate the model
predictions <- predict(rf_model, newdata = test_data)

summary(predictions)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    12.87   25.44   36.39   36.34   44.84  114.24

#predictions

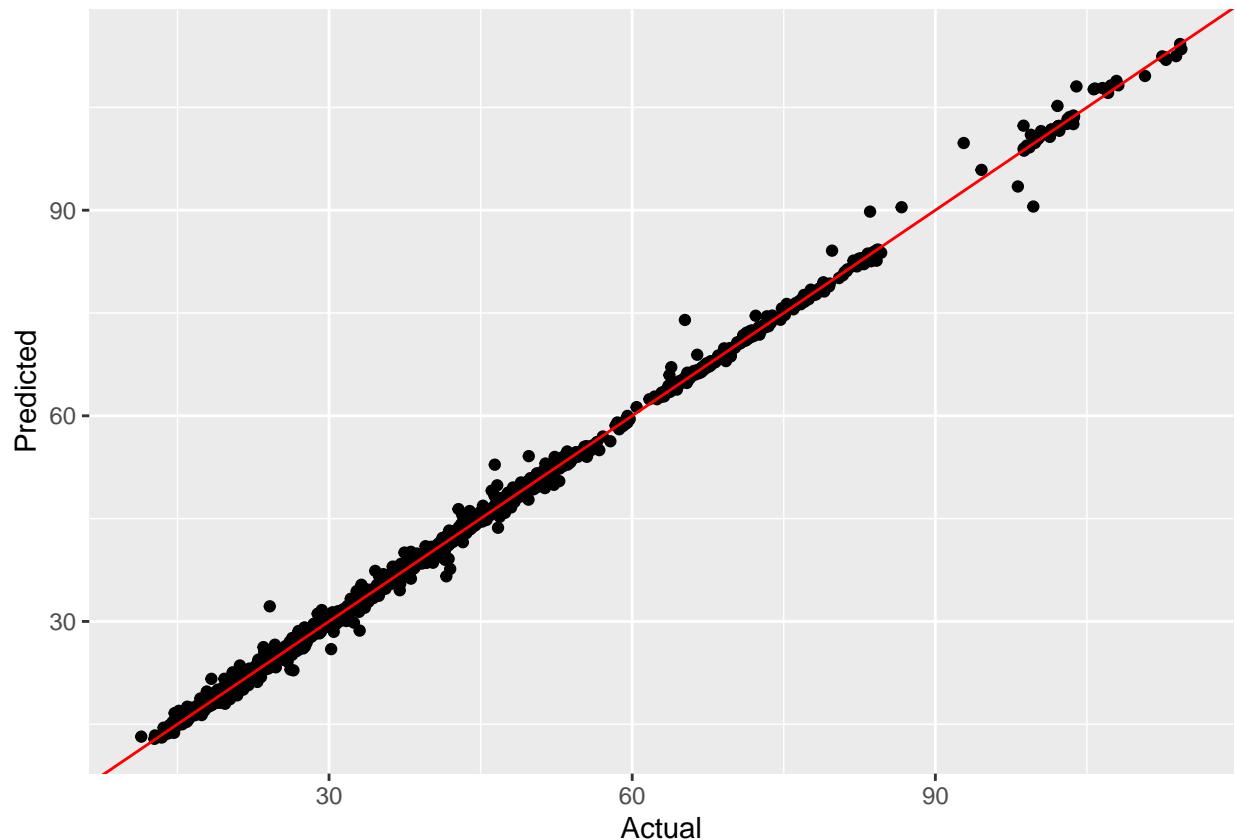
```

we are using ggplot2 to create a scatter plot of the actual vs. predicted values on the testing set. We are also adding a diagonal dashed line to represent perfect predictions.

```

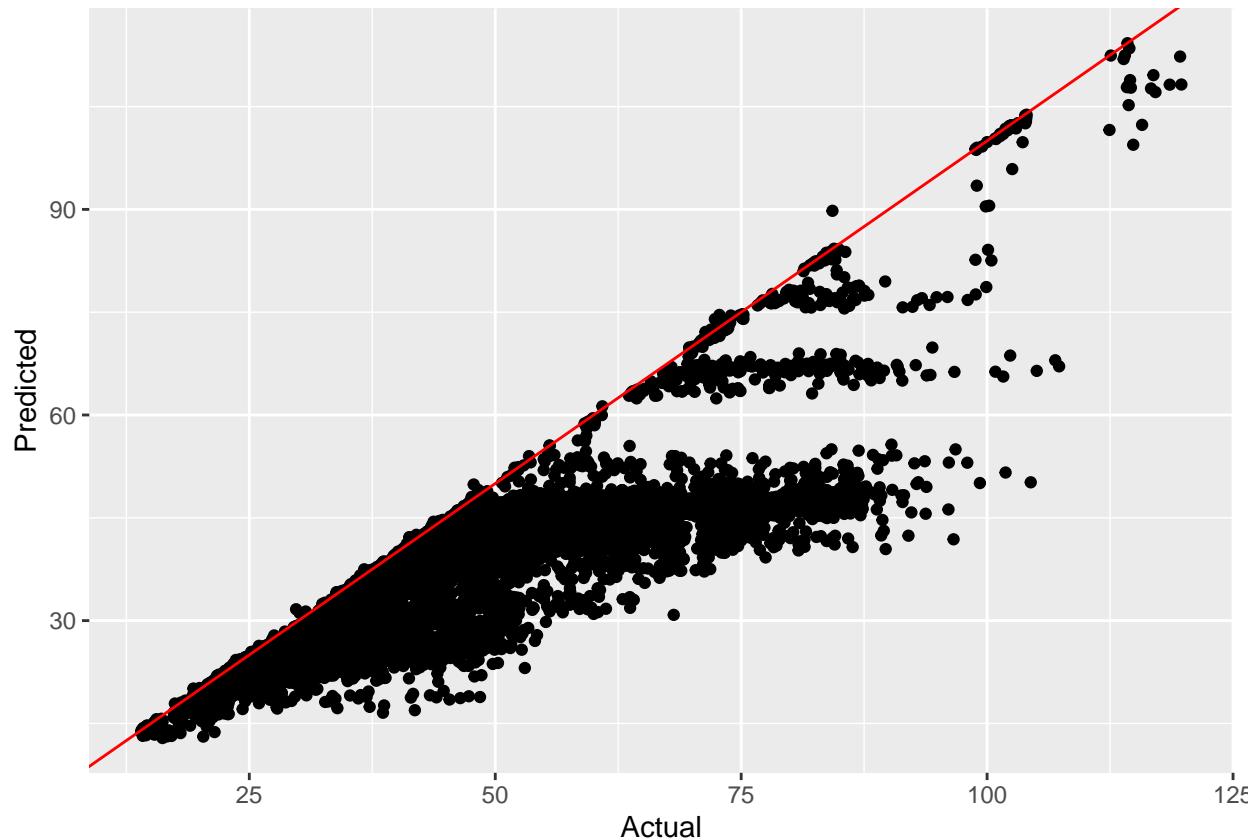
# To predict the trends of Fc with the test data
ggplot(data.frame(Predicted=predictions, Actual=test_data$Fc), aes(x=Actual, y=Predicted)) +
  geom_point() +
  geom_abline(intercept=0, slope=1, color="red")

```



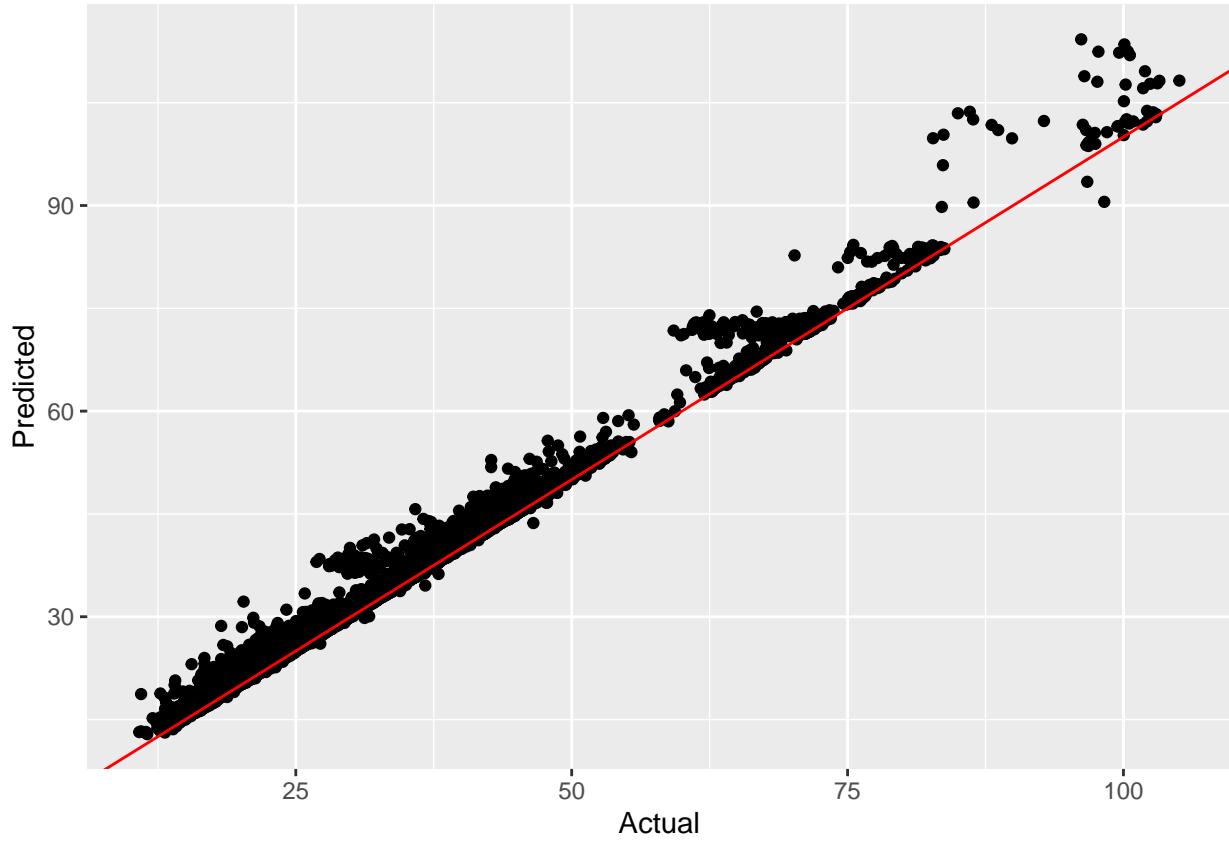
This scatterplot of the predicted values on the y-axis and actual values on the x-axis, with a red line showing perfect agreement between the two. The result: Fc (Frequency of the call) attribute is perfectly fit in the test data

```
# To predict the trends of HiFreq with the test data
ggplot(data.frame(Predicted=predictions, Actual=test_data$HiFreq), aes(x=Actual, y=Predicted)) +
  geom_point() +
  geom_abline(intercept=0, slope=1, color="red")
```



Result: The HiFreq attribute performs lower fit in the test data

```
# To predict the trends of LowFreq with the test data
ggplot(data.frame(Predicted=predictions, Actual=test_data$LowFreq),
        aes(x=Actual, y=Predicted)) +
  geom_point() +
  geom_abline(intercept=0, slope=1, color="red")
```



Result: The LowFreq data trend performs higher fit in the test data

```
predictions <- as.factor(predictions)
test_data$Fc <- as.factor(test_data$Fc)
```

we are using the predict function to make predictions on the testing set using the trained random forest model. Then, we are using the confusionMatrix function from the caret package to evaluate the performance of the model. This function provides a table of true positive, true negative, false positive, and false negative counts, as well as various performance metrics such as accuracy, sensitivity, and specificity.

```
#confusionMatrix(predictions, test_data$Fc) %>%
#  ggplot(aes(x = Prediction, y = Reference, fill = log10(Frequency))) +
#  geom_tile() +
#  scale_fill_gradient(low = "white", high = "red") +
#  labs(x = "Predicted", y = "Actual")
# Error: `data` and `reference` should be factors with the same levels.

# Reserved code for re-run to create the confusion matrix
# Convert predicted values to factor
#predictions <- factor(predictions, levels = levels(test_data$Fc))
# Convert actual values to factor
#actual <- factor(test_data$Fc, levels = levels(predictions))
# Create confusion matrix
#caret::confusionMatrix(predictions, actual)
```

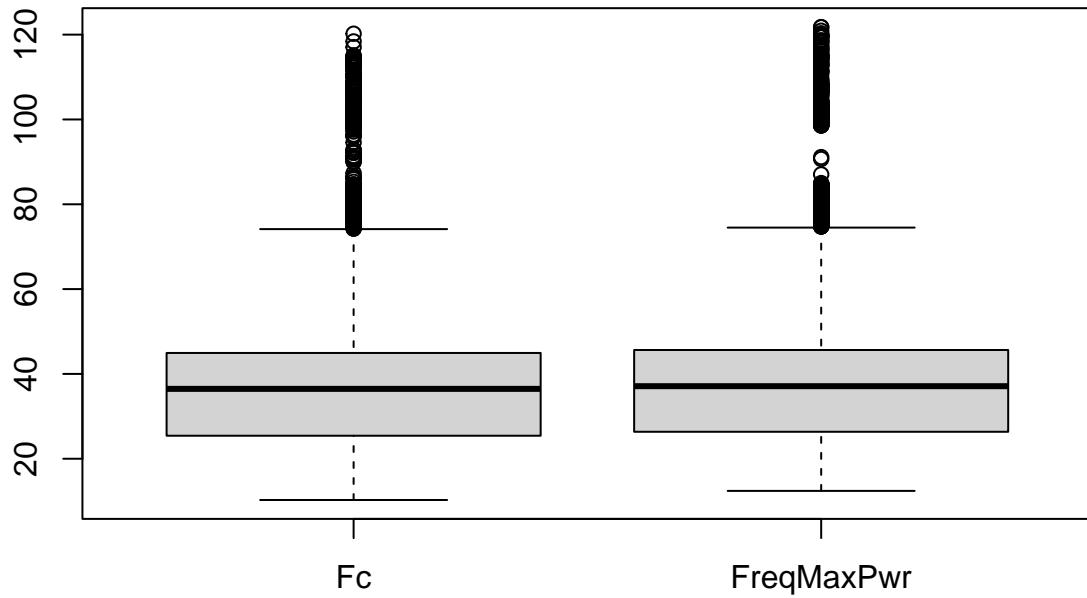
Note: confusionMatrix() is a function in R used to evaluate the performance of a classification model. It takes in two vectors, one with the predicted classes and the other with the actual classes, and produces a

confusion matrix that summarizes the results. From the confusion matrix, various performance metrics such as accuracy, precision, recall, and F1 score can be calculated to assess the model's performance. `confusionMatrix()` is particularly useful for multi-class classification problems, where it can handle more than two classes and produce metrics that take into account the imbalances in class distribution.

```
# Check the correlation value between two variables
cor(df22$Fc, df22$FreqMaxPwr)
```

```
## [1] 0.9930315
```

```
df22 |>
  select(Fc, FreqMaxPwr) |>
  boxplot()
```



MODEL SELECTION - MARS

MARS (Multivariate Adaptive Regression Splines) is a type of regression analysis that is designed to be flexible in handling both linear and nonlinear relationships between the response variable and the predictors.

One of the advantages of MARS is that it uses a stepwise approach to build the model, where the algorithm iteratively adds basis functions (splines) and interactions between them, based on their predictive power. This allows MARS to capture complex relationships between the predictors and the response variable, even in high-dimensional datasets.

Because MARS is built using a stepwise approach that selects only the most relevant predictors and interactions, it has a built-in method for variable selection and regularization. This means that MARS can avoid

overfitting the training data, making it less prone to the problem of overfitting that typically arises when using other machine learning models with many parameters.

Therefore, MARS may not necessarily need a separate train/test split, as it automatically selects the best predictors and interactions during the modeling process. However, it is still recommended to use some form of cross-validation or resampling technique to validate the model's performance and ensure that it is generalizable to new data.

```
# fit a basic MARS model - Multivariate Adaptive Regression Splines
mars1 <- earth(
  Fc ~ ., data = df22_new
)
summary(mars1)

## Call: earth(formula=Fc~., data=df22_new)
##
##                               coefficients
## (Intercept)                  101.191645
## h(12.062-Bndwdth)            0.084999
## h(Bndwdth-12.062)           -0.033662
## h(-0.364327-HiFtoFcExpAmp)  0.732087
## h(HiFtoFcExpAmp- -0.364327) -1.490753
## h(101.002-FreqCtr)          -1.010274
## h(FreqCtr-101.002)           1.085139
##
## Selected 7 of 7 terms, and 3 of 101 predictors
## Termination condition: RSq changed by less than 0.001 at 7 terms
## Importance: FreqCtr, HiFtoFcExpAmp, Bndwdth, TimeInFile-unused, ...
## Number of terms at each degree of interaction: 1 6 (additive model)
## GCV 0.2868884    RSS 22142.33    GRSq 0.9981652    RSq 0.9981658
```

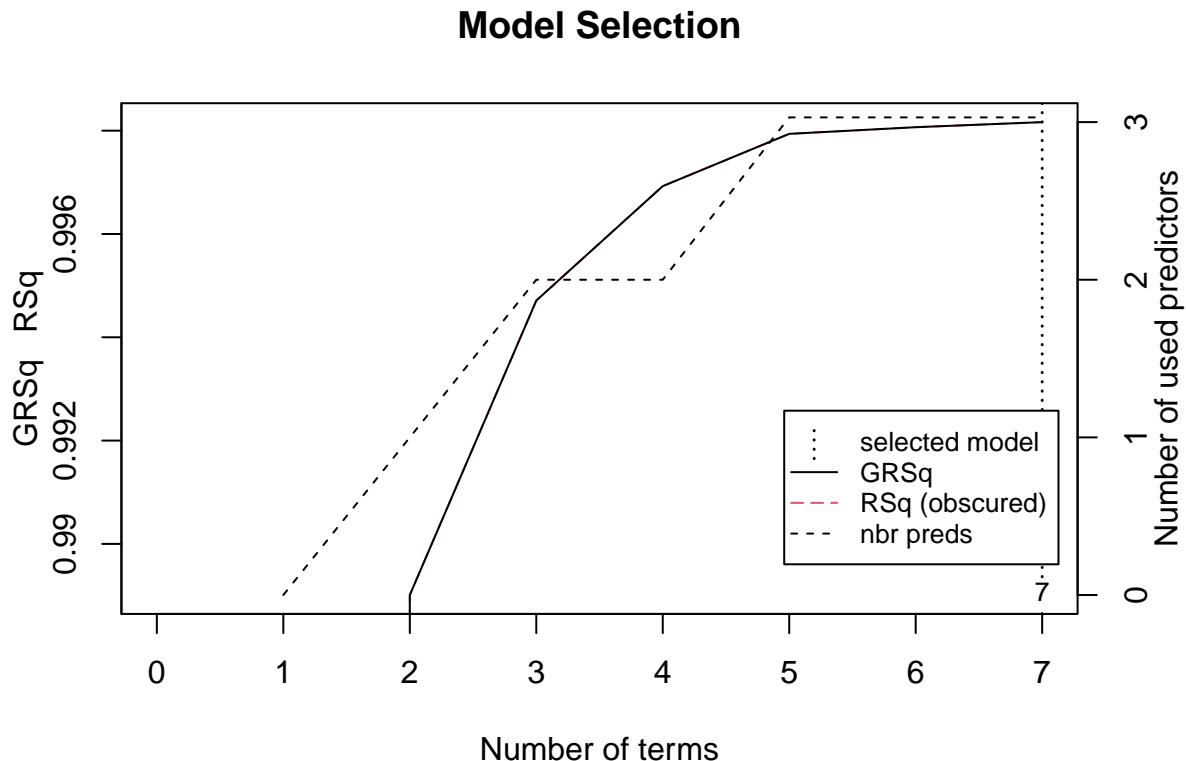
The `earth` function fits a multivariate adaptive regression spline (MARS) model to the data, which can capture nonlinear relationships and interactions between variables. In this case, the model appears to have selected seven terms to include in the final model, out of a total of 101 predictors.

The “Selected 7 of 7 terms, and 3 of 101 predictors” line indicates that the model has used only seven of the seven terms that were available in the formula you specified (i.e., all variables except for the response variable `Fc`). This suggests that some of the terms may have been redundant or not useful for predicting `Fc`.

The “Termination condition” line shows that the model stopped adding terms once the maximum R-squared value was reached, indicating that the model has a good fit to the data. The “Importance” line shows which predictors were most important in the model, with the `FreqCtr`, `HiFtoFcExpAmp`, and `Bndwdth`, and `TimeInFile` + `PrecedingIntrvl` variables apparently not contributing much to the fit.

- $\text{GCV} = 0.2868884$: Generalized Cross-Validation (GCV) score for the final model, where 0 is the minimum value and lower values indicate better models.
- $\text{RSS} = 22142.33$: Residual Sum of Squares (RSS) for the final model, which measures the total difference between the predicted and actual values of the response variable. A lower RSS score indicates a better fit of the model.
- $\text{GRSq} = 0.9981652$: Generalized R-Squared (GRSq) for the final model, which is a measure of how well the model fits the data, where 1 is the maximum value and higher values indicate better models.
- $\text{RSq} = 0.9981658$: R-Squared (RSq) for the final model, which is a similar measure of how well the model fits the data as GRSp. A higher R-squared score indicates a better fit of the model.

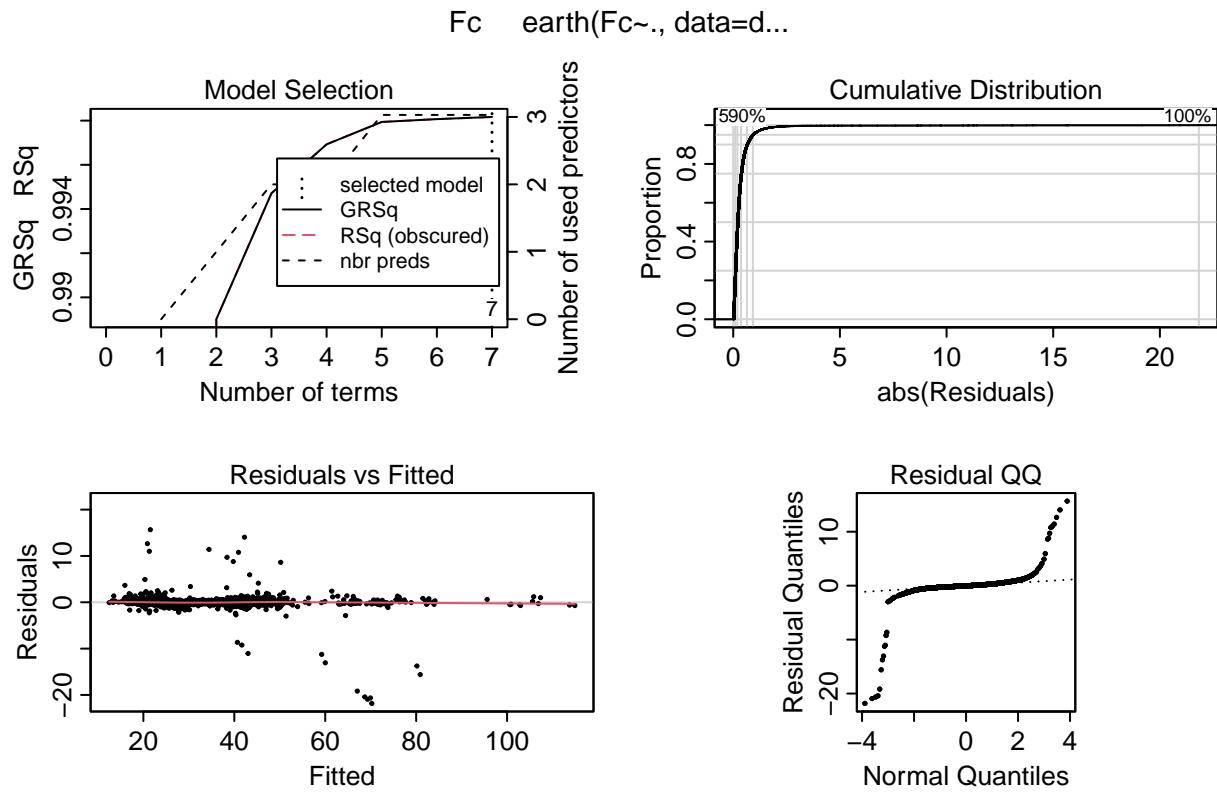
```
plot(mars1, which = 1)
```



The plot method for MARS model objects provide useful performance and residual plots. This model selection plot that graphs the GCV R² (left-hand y-axis and solid black line) based on the number of terms retained in the model (x-axis) which are constructed from a certain number of original predictors (right-hand y-axis). The vertical dashed line at 7 tells us the optimal number of non-intercept terms retained where marginal increases in GCV R² are less than 0.001.

For this model, 6 non-intercept terms were retained which are based on 3 predictors. Any additional terms retained in the model, over and above these 6, result in less than 0.001 improvement in the GCV R².

```
# Other views in plot of model MARS1  
plot(mars1)
```



MARS Model degree 2

```
# fit a basic MARS model - Multivariate Adaptive Regression Splines
mars2 <- earth(
  Fc ~ ., data = df22_new,
  degree = 2
)
```

```
summary(mars2)
```

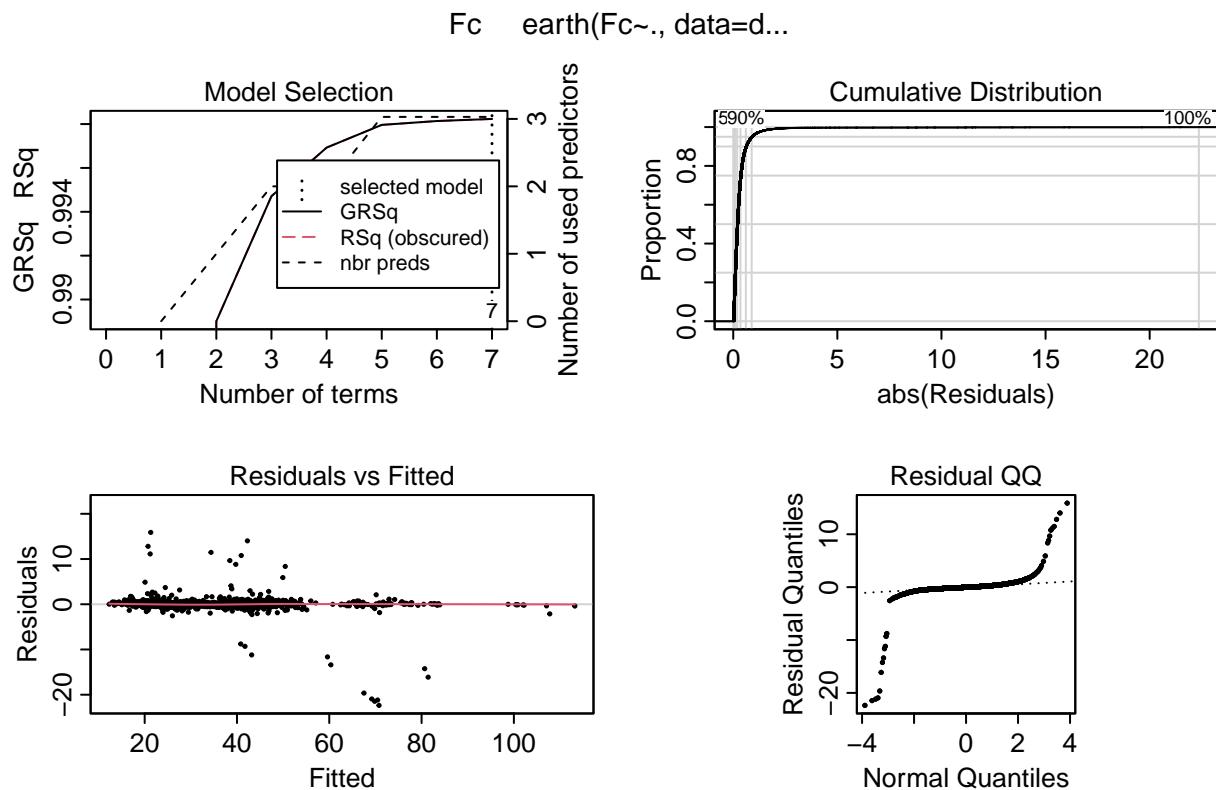
```
## Call: earth(formula=Fc~, data=df22_new, degree=2)
##
##                               coefficients
## (Intercept)                  101.714287
## h(-0.364327-HiFtoFcExpAmp)    0.723449
## h(HiFtoFcExpAmp- -0.364327)   -1.482365
## h(101.002-FreqCtr)            -1.018904
## h(FreqCtr-101.002)             1.035273
## h(12.2588-Bndwdth) * h(101.002-FreqCtr)  0.001342
## h(Bndwdth-12.2588) * h(101.002-FreqCtr) -0.000715
##
## Selected 7 of 7 terms, and 3 of 101 predictors
## Termination condition: RSq changed by less than 0.001 at 7 terms
```

```

## Importance: FreqCtr, HiFtoFcExpAmp, Bndwdth, TimeInFile-unused, ...
## Number of terms at each degree of interaction: 1 4 2
## GCV 0.2760953    RSS 21307.66    GRSq 0.9982342    RSq 0.9982349

plot(mars2)

```



MARS model with CallDuration as a predictor

```

# fit a basic MARS model - Multivariate Adaptive Regression Splines
mars3 <- earth(
  CallDuration ~ ., data = df22_new
)
summary(mars3)

```

```

## Call: earth(formula=CallDuration~., data=df22_new)
##
##                               coefficients
## (Intercept)                  -58.473564
## h(89.6449-PrcntMaxAmpDur)   -0.056634
## h(PrcntMaxAmpDur-89.6449)   -0.073052
## h(0.072643-TimeFromMaxToFc) 0.320816
## h(TimeFromMaxToFc-0.072643)  0.713869
## h(TimeFromMaxToFc-4.86627)   -0.280562

```

```

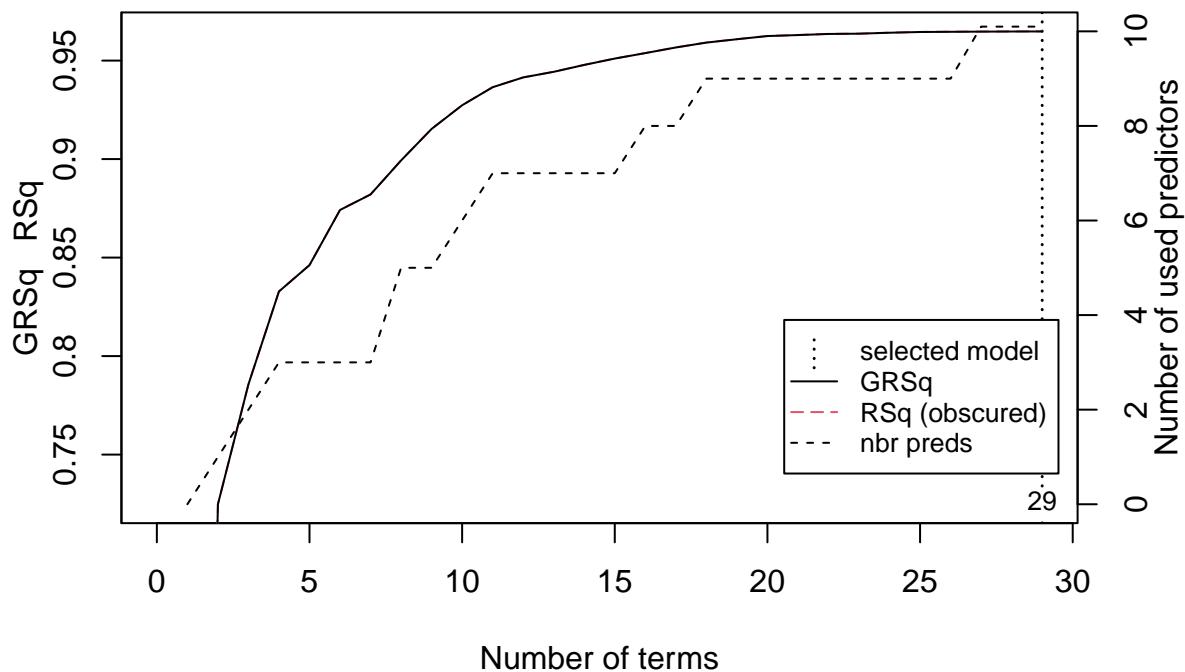
## h(TimeFromMaxToFc-16.1513)      -0.321264
## h(22.8773-PrcntKneeDur)        -0.016613
## h(PrcntKneeDur-22.8773)        0.073163
## h(PrcntKneeDur-79.4364)        0.244038
## h(-0.016129-EndSlope)         -0.170821
## h(EndSlope- -0.016129)         -0.151106
## h(0.740731-HiFtoKnDmp)        2.045418
## h(HiFtoKnDmp-0.740731)        0.103476
## h(1.38456-LedgeDuration)       0.179175
## h(LedgeDuration-1.38456)       0.077828
## h(KnToFcDur-1.37676)          -2.056078
## h(4.40175-KnToFcDur)          -3.106349
## h(KnToFcDur-4.40175)          2.875646
## h(KnToFcDur-13.3944)          -0.584730
## h(Bndw5dB-0.294253)          54.915279
## h(Bndw5dB-0.487221)          33.322183
## h(1.13947-Bndw5dB)           89.432982
## h(Bndw5dB-1.13947)            -88.319039
## h(2.61476-DurOf15dB)          0.186994
## h(DurOf15dB-2.61476)          0.174106
## h(FcMinusEndF-0)              1.594257
## h(3.94841-FcMinusEndF)        0.592041
## h(FcMinusEndF-3.94841)        -1.416935
##
## Selected 29 of 29 terms, and 10 of 101 predictors
## Termination condition: RSq changed by less than 0.001 at 29 terms
## Importance: TimeFromMaxToFc, DurOf15dB, PrcntMaxAmpDur, KnToFcDur, Bndw5dB, ...
## Number of terms at each degree of interaction: 1 28 (additive model)
## GCV 1.474584    RSS 113680.2    GRSq 0.9647895   RSq 0.9648405

```

This model is good as well but not better than model 1 and model 2 based on these scores: GCV, RSS, GRSq, and RSq

```
plot(mars3, which = 1)
```

Model Selection



```
plot(mars3)
```

CallDuration earth(Ca...)

