Project

2024-11-16

```r
install.packages("ucimlrepo")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
install.packages("nnet")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
library(nnet)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(ucimlrepo)
```

```r
# Fetch the dataset
land_mines <- fetch_ucirepo(id = 763)
```

```r
names(land_mines)
```

```
## [1] "data"      "metadata"  "variables"
```

```r
str(land_mines)
```

```
## List of 3
##  $ data     :List of 5
##   ..$ ids     : NULL
##   ..$ features:'data.frame': 338 obs. of  3 variables:
##   .. ..$ V: num [1:338] 0.338 0.32 0.287 0.256 0.263 ...
##   .. ..$ H: num [1:338] 0 0.182 0.273 0.455 0.545 ...
##   .. ..$ S: num [1:338] 0 0 0 0 0 0 0 0 0.6 0.6 ...
##   ..$ targets :'data.frame': 338 obs. of  1 variable:
##   .. ..$ M: int [1:338] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ original:'data.frame': 338 obs. of  4 variables:
##   .. ..$ V: num [1:338] 0.338 0.32 0.287 0.256 0.263 ...
```

```
##   .. ..$ H: num [1:338] 0 0.182 0.273 0.455 0.545 ...
##   .. ..$ S: num [1:338] 0 0 0 0 0 0 0 0 0.6 0.6 ...
##   .. ..$ M: int [1:338] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ headers : chr [1:4] "V" "H" "S" "M"
##  $ metadata :List of 22
##   ..$ uci_id                 : int 763
##   ..$ name                   : chr "Land Mines"
##   ..$ repository_url         : chr "https://archive.ics.uci.edu/dataset/763/land+mines-1"
##   ..$ data_url               : chr "https://archive.ics.uci.edu/static/public/763/data.csv"
##   ..$ abstract               : chr "Detection of mines buried in the ground is very important in te
##   ..$ area                   : chr "Engineering"
##   ..$ tasks                  :List of 2
##   .. ..$ : chr "Classification"
##   .. ..$ : chr "Clustering"
##   ..$ characteristics        :List of 3
##   .. ..$ : chr "Tabular"
##   .. ..$ : chr "Multivariate"
##   .. ..$ : chr "Other"
##   ..$ num_instances          : int 338
##   ..$ num_features           : int 3
##   ..$ feature_types          :List of 2
##   .. ..$ : chr "Real"
##   .. ..$ : chr "Integer"
##   ..$ demographics           : list()
##   ..$ target_col             :List of 1
##   .. ..$ : chr "M"
##   ..$ index_col              : NULL
##   ..$ has_missing_values     : chr "no"
##   ..$ missing_values_symbol  : NULL
##   ..$ year_of_dataset_creation: int 2018
##   ..$ last_updated           : chr "Thu Mar 07 2024"
##   ..$ dataset_doi            : chr "10.24432/C54C8Z"
##   ..$ creators               :List of 1
##   .. ..$ : chr "Hamdi Tolga KAHRAMAN"
##   ..$ intro_paper            :List of 16
##   .. ..$ ID     : int 163
##   .. ..$ type   : chr "NATIVE"
##   .. ..$ title  : chr "Passive Mine Detection and Classification Method Based on Hybrid Model"
##   .. ..$ authors: chr "C. Yilmaz, H. Kahraman, Salih Söyler"
##   .. ..$ venue  : chr "IEEE Access"
##   .. ..$ year   : int 2018
##   .. ..$ journal: chr "IEEE Access"
##   .. ..$ DOI    : NULL
##   .. ..$ URL    : chr "https://www.semanticscholar.org/paper/c70a13956cfeb634a235bc9a5b3398f2000fe68
##   .. ..$ sha    : NULL
##   .. ..$ corpus : NULL
##   .. ..$ arxiv  : NULL
##   .. ..$ mag    : NULL
##   .. ..$ acl    : NULL
##   .. ..$ pmid   : NULL
##   .. ..$ pmcid  : NULL
##   ..$ additional_info        :List of 9
##   .. ..$ summary                 : chr "Yilmaz, C., Kahraman, H. T., & Söyler, S. (2018). Passive mi
##   .. ..$ purpose                 : chr "PhD thesis"
```

```
##   .. ..$ funded_by                : chr "Cemal YILMAZ\nDepartment of Electrical and Electronics Engi
##   .. ..$ instances_represent      : chr "Land Mines"
##   .. ..$ recommended_data_splits  : chr "no"
##   .. ..$ sensitive_data           : chr "No"
##   .. ..$ preprocessing_description: chr "processing of missing"
##   .. ..$ variable_info            : NULL
##   .. ..$ citation                 : chr "Yilmaz, C., Kahraman, H. T., & Söyler, S. (2018). Passive mi
##  $ variables:'data.frame':   4 obs. of  7 variables:
##   ..$ name          : chr [1:4] "V" "H" "S" "M"
##   ..$ role          : chr [1:4] "Feature" "Feature" "Feature" "Target"
##   ..$ type          : chr [1:4] "Continuous" "Continuous" "Continuous" "Integer"
##   ..$ demographic   : logi [1:4] NA NA NA NA
##   ..$ description   : chr [1:4] "voltage: output voltage value of FLC sensor due to magnetic distort
##   ..$ units         : chr [1:4] "V" "cm" NA NA
##   ..$ missing_values: chr [1:4] "no" "no" "no" "no"
```

```r
land_mines_df <- land_mines$data$original
head(land_mines_df)
```

```
##           V         H S M
## 1 0.3381568 0.0000000 0 1
## 2 0.3202413 0.1818182 0 1
## 3 0.2870087 0.2727273 0 1
## 4 0.2562836 0.4545455 0 1
## 5 0.2628396 0.5454545 0 1
## 6 0.2409665 0.7272727 0 1
```

```r
# Data (as data frames)
X <- land_mines$data$features
y <- land_mines$data$targets
```

```r
# Metadata
print(land_mines$metadata)
```

```
## $uci_id
## [1] 763
##
## $name
## [1] "Land Mines"
##
## $repository_url
## [1] "https://archive.ics.uci.edu/dataset/763/land+mines-1"
##
## $data_url
## [1] "https://archive.ics.uci.edu/static/public/763/data.csv"
##
## $abstract
## [1] "Detection of mines buried in the ground is very important in terms of safety of life and propert
##
## $area
## [1] "Engineering"
##
## $tasks
## $tasks[[1]]
## [1] "Classification"
##
```

```
## $tasks[[2]]
## [1] "Clustering"
##
##
## $characteristics
## $characteristics[[1]]
## [1] "Tabular"
##
## $characteristics[[2]]
## [1] "Multivariate"
##
## $characteristics[[3]]
## [1] "Other"
##
##
## $num_instances
## [1] 338
##
## $num_features
## [1] 3
##
## $feature_types
## $feature_types[[1]]
## [1] "Real"
##
## $feature_types[[2]]
## [1] "Integer"
##
##
## $demographics
## list()
##
## $target_col
## $target_col[[1]]
## [1] "M"
##
##
## $index_col
## NULL
##
## $has_missing_values
## [1] "no"
##
## $missing_values_symbol
## NULL
##
## $year_of_dataset_creation
## [1] 2018
##
## $last_updated
## [1] "Thu Mar 07 2024"
##
## $dataset_doi
## [1] "10.24432/C54C8Z"
```

```
##
## $creators
## $creators[[1]]
## [1] "Hamdi Tolga KAHRAMAN"
##
##
## $intro_paper
## $intro_paper$ID
## [1] 163
##
## $intro_paper$type
## [1] "NATIVE"
##
## $intro_paper$title
## [1] "Passive Mine Detection and Classification Method Based on Hybrid Model"
##
## $intro_paper$authors
## [1] "C. Yilmaz, H. Kahraman, Salih Söyler"
##
## $intro_paper$venue
## [1] "IEEE Access"
##
## $intro_paper$year
## [1] 2018
##
## $intro_paper$journal
## [1] "IEEE Access"
##
## $intro_paper$DOI
## NULL
##
## $intro_paper$URL
## [1] "https://www.semanticscholar.org/paper/c70a13956cfeb634a235bc9a5b3398f2000fe68c"
##
## $intro_paper$sha
## NULL
##
## $intro_paper$corpus
## NULL
##
## $intro_paper$arxiv
## NULL
##
## $intro_paper$mag
## NULL
##
## $intro_paper$acl
## NULL
##
## $intro_paper$pmid
## NULL
##
## $intro_paper$pmcid
## NULL
```

```
##
##
## $additional_info
## $additional_info$summary
## [1] "Yilmaz, C., Kahraman, H. T., & Söyler, S. (2018). Passive mine detection and classification met
##
## $additional_info$purpose
## [1] "PhD thesis"
##
## $additional_info$funded_by
## [1] "Cemal YILMAZ\nDepartment of Electrical and Electronics Engineering, Gazi University, Ankara, Tu
##
## $additional_info$instances_represent
## [1] "Land Mines"
##
## $additional_info$recommended_data_splits
## [1] "no"
##
## $additional_info$sensitive_data
## [1] "No"
##
## $additional_info$preprocessing_description
## [1] "processing of missing"
##
## $additional_info$variable_info
## NULL
##
## $additional_info$citation
## [1] "Yilmaz, C., Kahraman, H. T., & Söyler, S. (2018). Passive mine detection and classification met
```

```r
# Variable information
print(land_mines$variables)
```

```
##   name    role       type demographic
## 1    V Feature Continuous          NA
## 2    H Feature Continuous          NA
## 3    S Feature Continuous          NA
## 4    M  Target    Integer          NA
##
## 1                                                                                              volt
## 2
## 3 soil type: 6 different soil types depending on the moisture condition [dry and sandy, dry and humu
## 4                                                                                      mine type: 
##   units missing_values
## 1     V             no
## 2    cm             no
## 3  <NA>             no
## 4  <NA>             no
```

```r
str(X)
```

```
## 'data.frame':    338 obs. of  3 variables:
##  $ V: num  0.338 0.32 0.287 0.256 0.263 ...
##  $ H: num  0 0.182 0.273 0.455 0.545 ...
##  $ S: num  0 0 0 0 0 0 0 0 0.6 0.6 ...
```

```r
str(y)
```

```
## 'data.frame':    338 obs. of  1 variable:
##  $ M: int  1 1 1 1 1 1 1 1 1 1 ...
```

```r
#Converting target variable to factor
y <- factor(y$M)
str(y)
```

```
##  Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
```

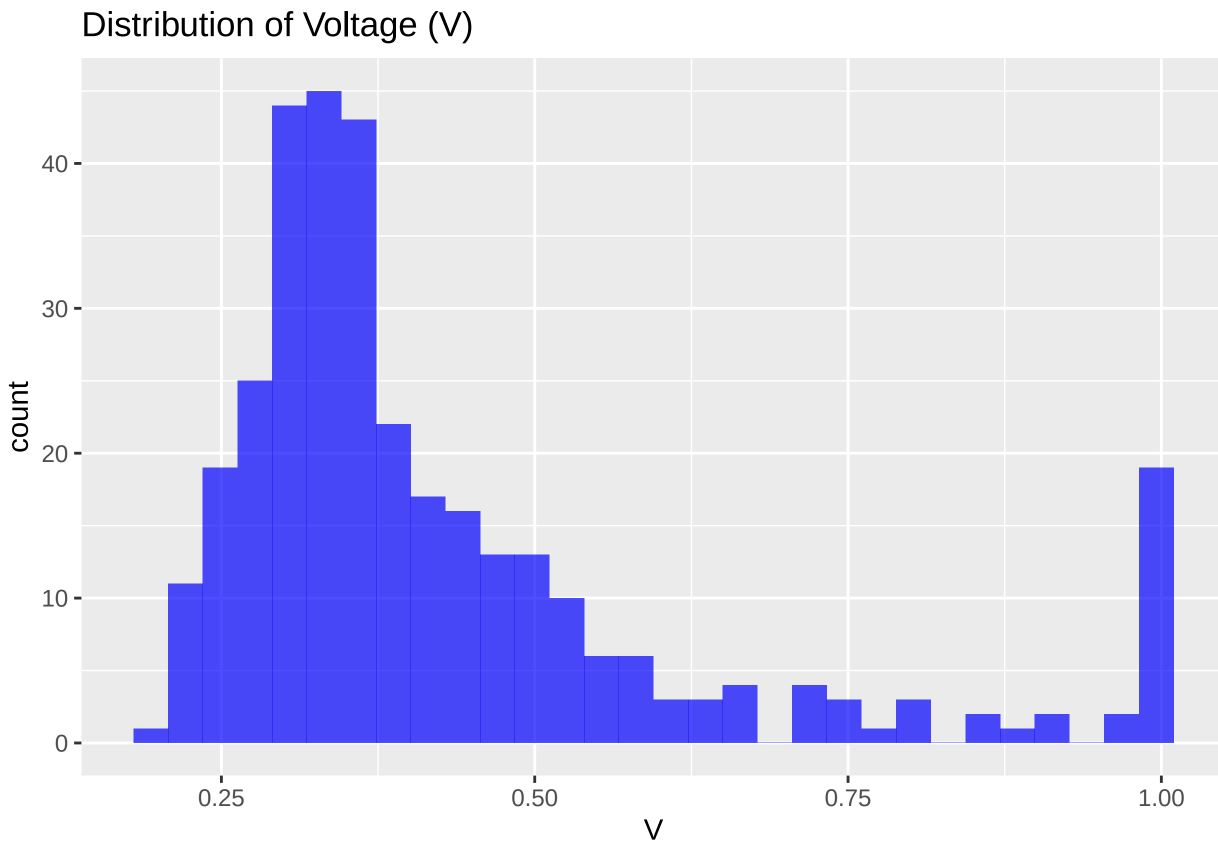
```r
# Finding duplicates in the features and target
duplicate_X <- sum(duplicated(land_mines$data$features))
duplicate_y <- sum(duplicated(land_mines$data$targets))
cat("Number of duplicate rows in X: ", duplicate_X, "\n", "Number of duplicate entries in y: ", duplica
```

```
## Number of duplicate rows in X:  0
##  Number of duplicate entries in y:  333
```

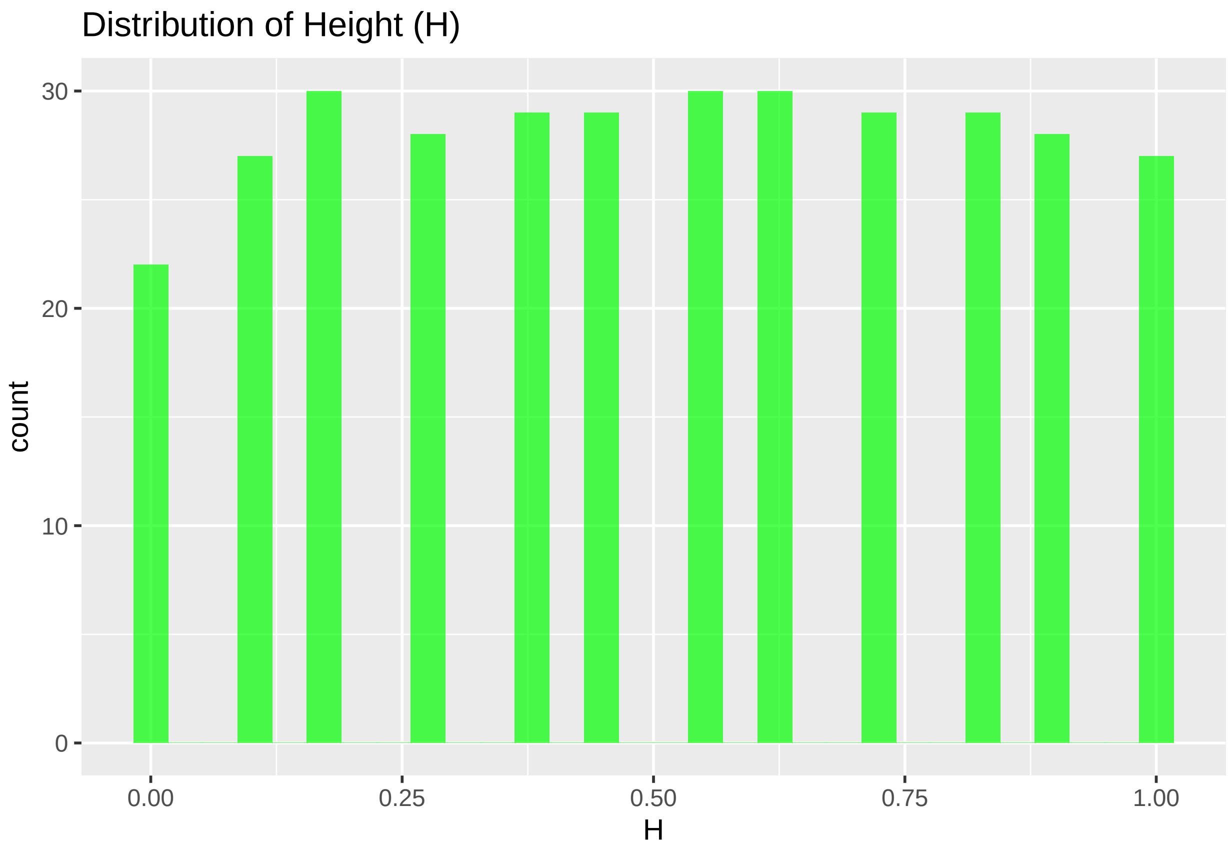
```r
#Summary of the features
summary(X)
```

```
##        V                H                S         
##  Min.   :0.1977   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.3097   1st Qu.:0.2727   1st Qu.:0.2000  
##  Median :0.3595   Median :0.5455   Median :0.6000  
##  Mean   :0.4306   Mean   :0.5089   Mean   :0.5036  
##  3rd Qu.:0.4826   3rd Qu.:0.7273   3rd Qu.:0.8000  
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
```

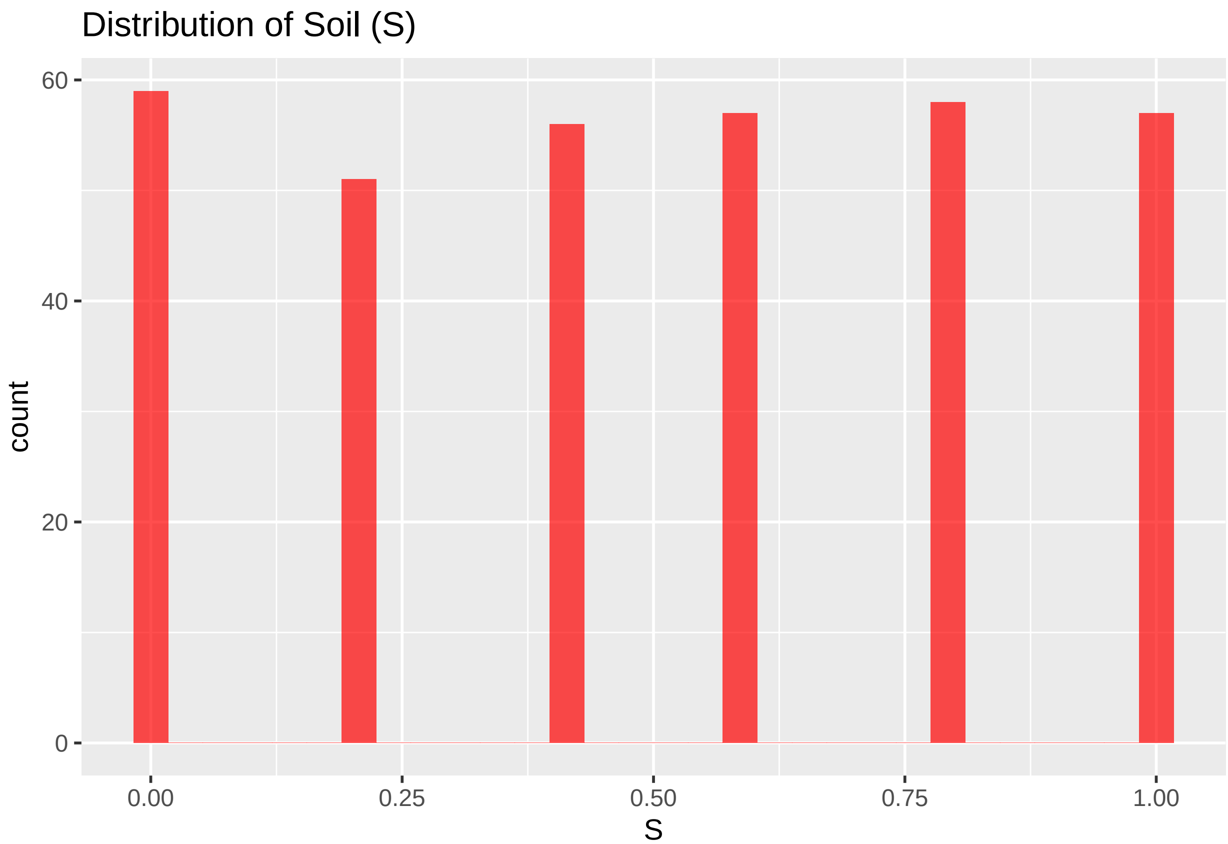
```r
# Visualizing distributions of Voltage
ggplot(X, aes(x = V)) + geom_histogram(bins = 30, fill = 'blue', alpha
= 0.7) + labs(title = "Distribution of Voltage (V)")
```




```r
# Visualising distributions of Height
ggplot(X, aes(x = H)) + geom_histogram(bins = 30, fill = 'green',
alpha = 0.7) + labs(title = "Distribution of Height (H)")
```

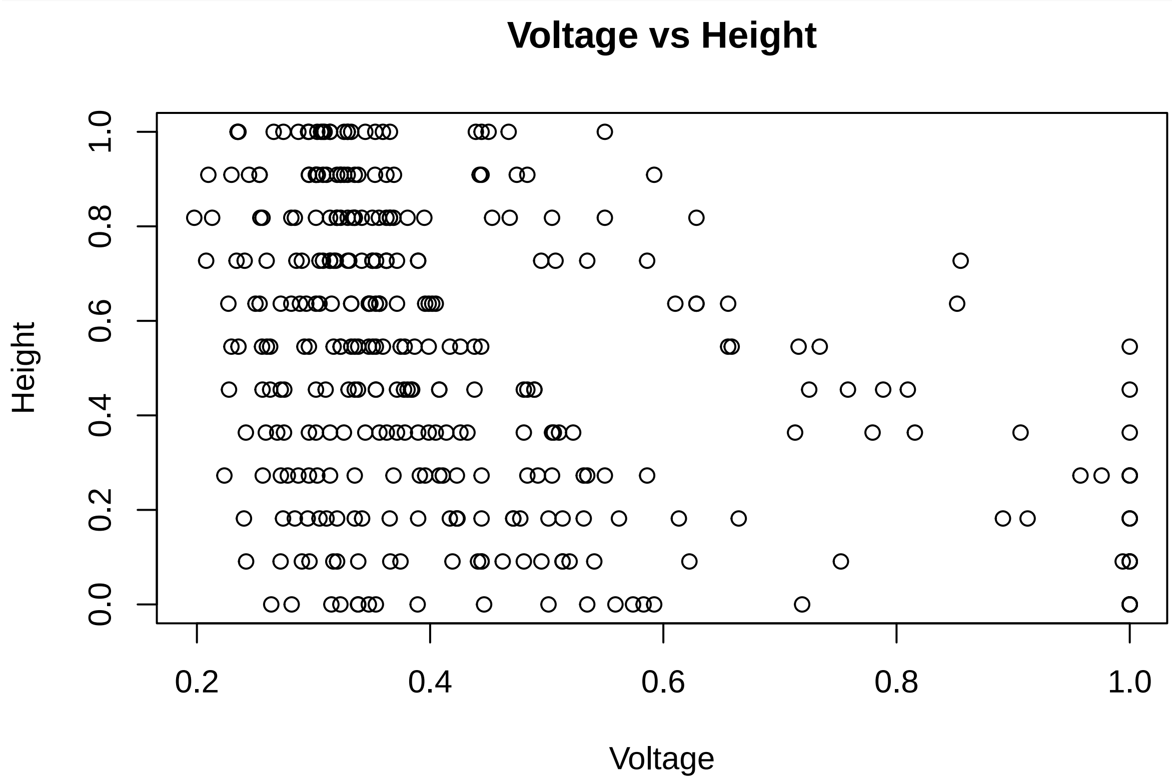



```r
# Visualising distributions of Soil
ggplot(X, aes(x = S)) + geom_histogram(bins = 30, fill = 'red', alpha
= 0.7) + labs(title = "Distribution of Soil (S)")
```

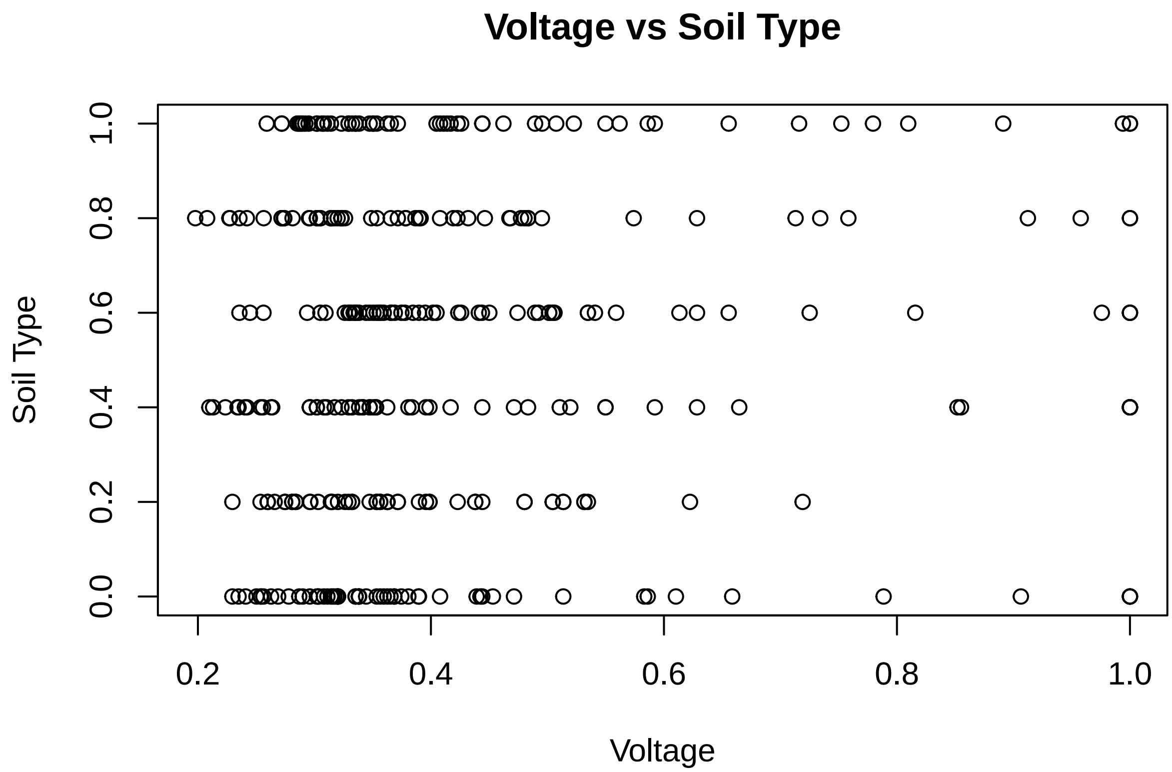



```r
# Distribution of mine type
barplot(table(land_mines_df$M), main = "Mine Type Distribution", xlab
= "Mine Type", ylab = "Frequency")
```

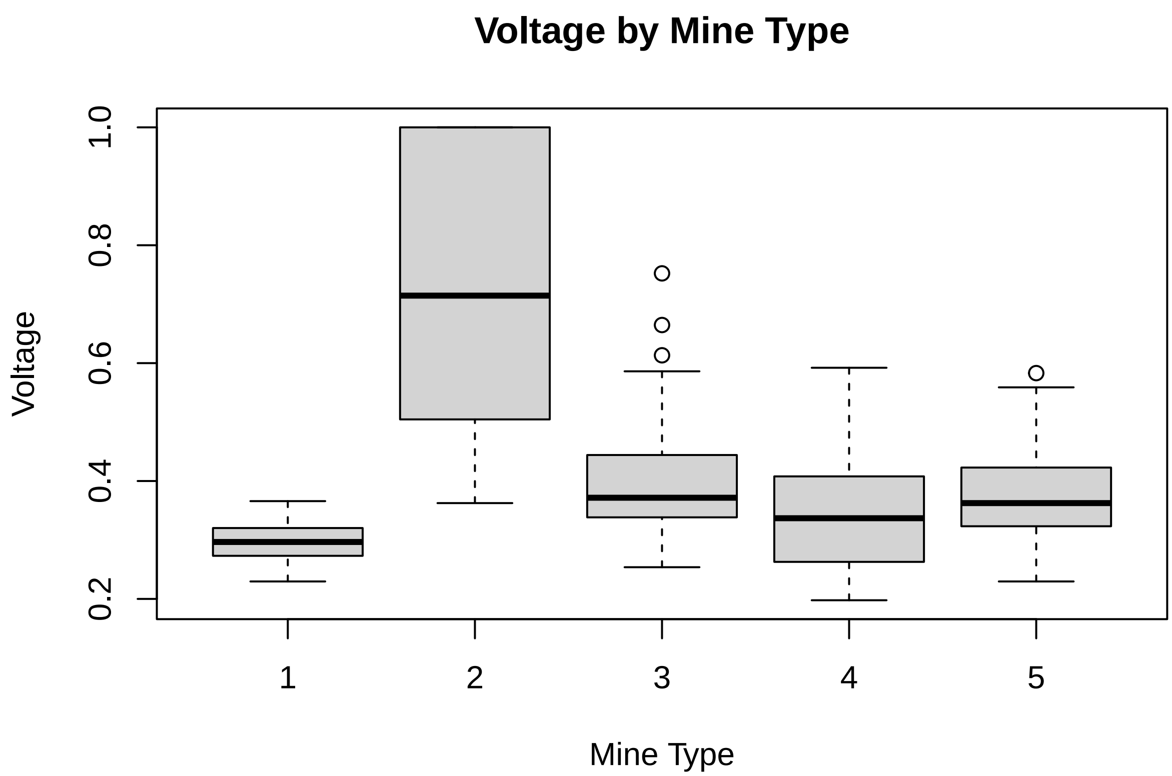

```r
# Plot of voltage vs height
plot(land_mines_df$V, land_mines_df$H, main = "Voltage vs Height",
xlab = "Voltage", ylab = "Height")
```



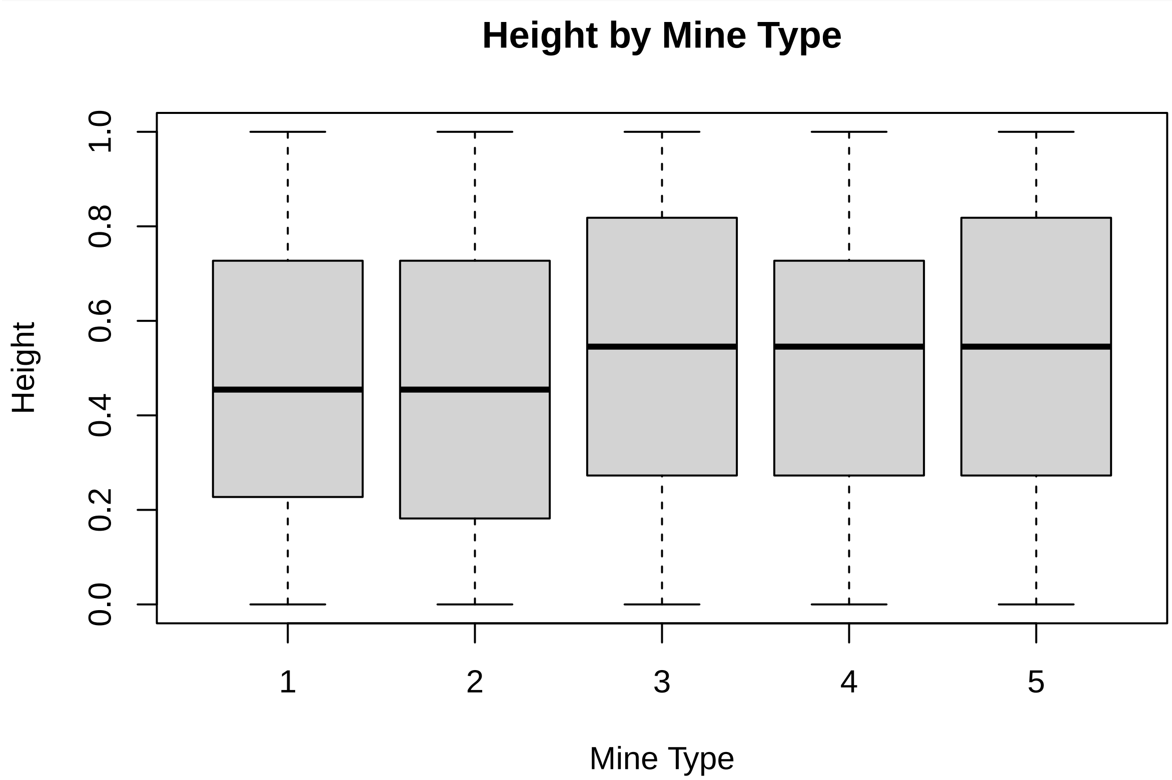
```r
# Plot of voltage vs soil type
plot(land_mines_df$V, land_mines_df$S, main = "Voltage vs Soil Type",
xlab = "Voltage", ylab = "Soil Type")
```




```r
# Plot of voltage by minetype
boxplot(V ~ M, data = land_mines_df, main = "Voltage by Mine Type",
xlab = "Mine Type", ylab = "Voltage")
```




```r
# Plot of height by minetype
boxplot(H ~ M, data = land_mines_df, main = "Height by Mine Type",
xlab = "Mine Type", ylab = "Height")
```



```r
# Apply PCA
X_scaled <- scale(X)
pca <- prcomp(X_scaled, center = TRUE, scale. = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3
## Standard deviation     1.1770 0.9987 0.7856
## Proportion of Variance 0.4618 0.3325 0.2057
## Cumulative Proportion  0.4618 0.7943 1.0000
```

```r
# K-means clustering model
set.seed(123)
kmeans_result <- kmeans(X_scaled, centers = 3, nstart = 25)
print(kmeans_result)
```

```
## K-means clustering with 3 clusters of sizes 144, 144, 50
##
## Cluster means:
##            V          H          S
## 1 -0.3014557  0.1710922  0.8692317
## 2 -0.3948505  0.1484011 -0.9140660
## 3  2.0053616 -0.9201408  0.1291228
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2
##  [38] 2 2 2 1 1 1 1 1 1 1 3 3 3 3 2 2 2 2 3 3 3 3 3 1 1 1 3 3 2 2 2 2 2 3 3 3 3
##  [75] 3 1 1 1 3 3 3 3 3 3 2 2 3 3 3 3 3 1 1 1 2 2 2 2 2 2 2 2 3 1 1 1 1 1 1 2 2
## [112] 2 2 2 2 2 3 1 1 1 1 1 1 1 3 2 2 2 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 1 1
## [149] 1 1 1 1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 1 1 1 1 1 1 1 3 2 2
## [186] 2 2 2 2 2 3 1 1 1 1 1 1 1 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 1 1 1
## [223] 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 3 3 2 2 3 3 1 1 2 2
## [260] 2 3 3 1 1 3 3 3 2 3 3 1 1 2 2 2 2 1 1 1 3 2 2 2 1 1 1 1 3 2 2 3 1 1 1 2 2
## [297] 2 2 1 1 1 2 2 2 2 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 3 1 1 1 2 2 1 1 1 1 3 2 2
## [334] 2 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 194.8586 195.5825 104.2797
##  (between_SS / total_SS =  51.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
# logistic regression model
data <- data.frame(X, y = as.factor(y))
model_multinom <- multinom(y ~ ., data = data)
```

```
## # weights:  25 (16 variable)
## initial  value 543.990014
## iter  10 value 411.174973
## iter  20 value 354.088472
## iter  30 value 353.431094
## final  value 353.427302
```

```
## converged
```

```r
summary(model_multinom)
```

```
## Call:
## multinom(formula = y ~ ., data = data)
##
## Coefficients:
##   (Intercept)        V         H          S
## 2  -32.771305 64.62394 13.504539 -3.7998918
## 3  -12.413425 31.03676  3.807539 -0.7349769
## 4   -5.547538 15.61813  1.148904 -0.1752905
## 5   -9.708252 25.07504  2.700693 -0.3493845
##
## Std. Errors:
##   (Intercept)        V         H         S
## 2    4.109183 7.610529 2.1051675 1.2949626
## 3    1.677159 4.096268 0.8511527 0.5984940
## 4    1.253255 3.453616 0.6502918 0.5223790
## 5    1.491917 3.824198 0.7613781 0.5681517
##
## Residual Deviance: 706.8546
## AIC: 738.8546
```

```r
# Knn modelling
set.seed(123)
train_indices <- createDataPartition(y, p = 0.8, list = FALSE)
train_data <- data.frame(X[train_indices, ], y = y[train_indices])
test_data <- data.frame(X[-train_indices, ], y = y[-train_indices])
train_control <- trainControl(method = "cv", number = 10)
k_value_tune <- expand.grid(k = seq(1, 20, by = 1))
# Training the k-NN model
knn_model <- train(y ~ .,
                   data = train_data,
                   method = "knn",
                   trControl = train_control,
                   tuneGrid = k_value_tune,
                   preProcess = c("center", "scale"))
print(knn_model)
```

```
## k-Nearest Neighbors
##
## 271 samples
##   3 predictor
##   5 classes: '1', '2', '3', '4', '5'
##
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 246, 242, 245, 243, 244, 244, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.4368121  0.2946143
##    2  0.4354798  0.2938798
##    3  0.4750041  0.3427135
```

```
##    4  0.4384792  0.2976271
##    5  0.3920050  0.2396782
##    6  0.3750254  0.2168534
##    7  0.3599965  0.1976595
##    8  0.3748624  0.2171614
##    9  0.3809518  0.2251727
##   10  0.4004710  0.2496163
##   11  0.4007866  0.2497683
##   12  0.4223951  0.2766792
##   13  0.3936976  0.2412349
##   14  0.4009404  0.2515474
##   15  0.3948396  0.2431880
##   16  0.4259780  0.2825400
##   17  0.3941772  0.2418686
##   18  0.4017476  0.2501618
##   19  0.3975335  0.2460078
##   20  0.3783414  0.2216260
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 3.
```

```r
print(paste("The Best k value for k-NN is", knn_model$bestTune$k))
```

```
## [1] "The Best k value for k-NN is 3"
```

```r
# Evaluate the model on the test set
knn_predictions <- predict(knn_model, newdata = test_data)
confusion_matrix_knn <- confusionMatrix(knn_predictions, test_data$y)

# Confusion matrix and accuracy
print(confusion_matrix_knn)
```

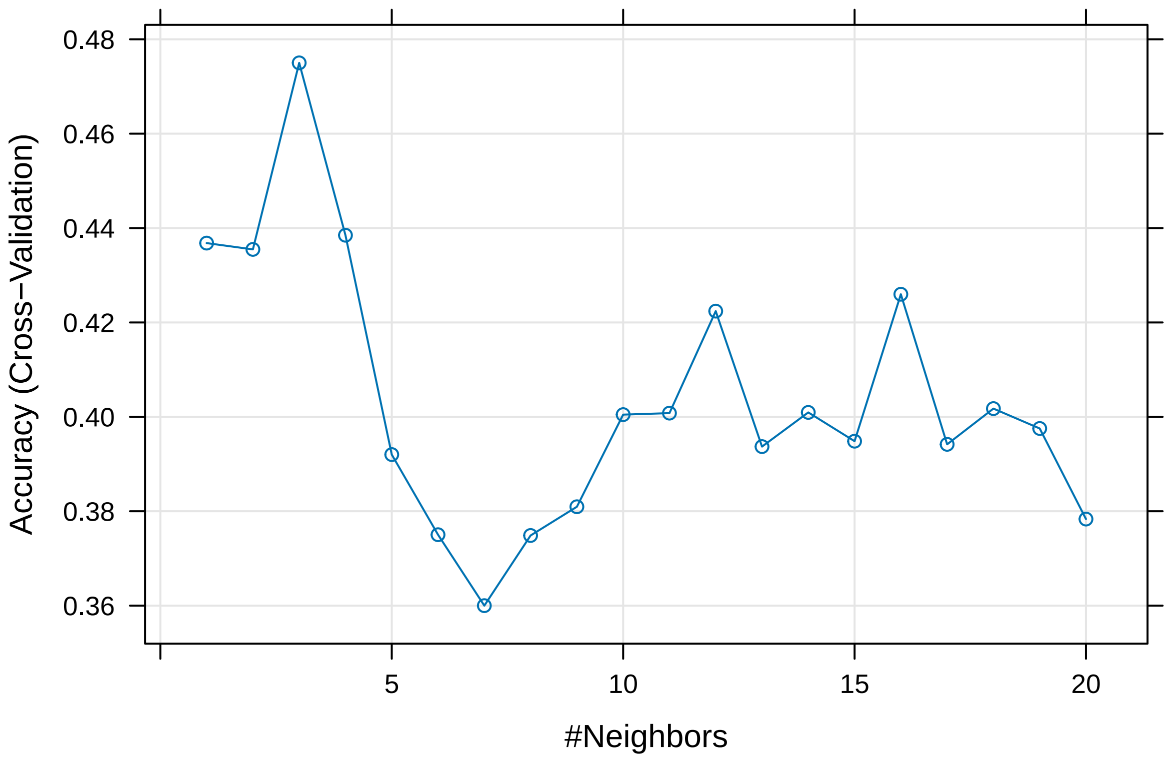
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5
##          1  7  0  1  1  1
##          2  0 11  1  0  0
##          3  1  0  5  3  2
##          4  1  3  2  6  5
##          5  5  0  4  3  5
##
## Overall Statistics
##
##                Accuracy : 0.5075
##                  95% CI : (0.3824, 0.6318)
##     No Information Rate : 0.209
##     P-Value [Acc > NIR] : 6.272e-08
##
##                   Kappa : 0.3852
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.5000   0.7857  0.38462  0.46154  0.38462
## Specificity            0.9434   0.9811  0.88889  0.79630  0.77778
## Pos Pred Value         0.7000   0.9167  0.45455  0.35294  0.29412
## Neg Pred Value         0.8772   0.9455  0.85714  0.86000  0.84000
## Prevalence             0.2090   0.2090  0.19403  0.19403  0.19403
## Detection Rate         0.1045   0.1642  0.07463  0.08955  0.07463
## Detection Prevalence   0.1493   0.1791  0.16418  0.25373  0.25373
## Balanced Accuracy      0.7217   0.8834  0.63675  0.62892  0.58120
```

```r
cat("The Accuracy of k-NN model:", round(confusion_matrix_knn$overall["Accuracy"] * 100, 2), "%\n")
```

```
## The Accuracy of k-NN model: 50.75 %
```

```r
# Plot k-NN model performance during tuning
plot(knn_model)
```




```r
# Random Forest model
model_rf <- randomForest(y ~ V + H + S, data = data, ntree = 100)
print(model_rf)
```

```
##
## Call:
##  randomForest(formula = y ~ V + H + S, data = data, ntree = 100)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 52.37%
## Confusion matrix:
##    1  2  3  4  5 class.error
## 1 54  0  6  3  8   0.2394366
```

```
## 2   0 60  1  9  0   0.1428571
## 3   6  3 19  8 30   0.7121212
## 4  11  5 14 19 17   0.7121212
## 5  12  1 31 12  9   0.8615385
```

```r
predictions <- predict(model_rf, data)
confusion_matrix <- table(predictions, data$y)
print(confusion_matrix)
```

```
##
## predictions  1  2  3  4  5
##           1 71  0  2  0  4
##           2  0 70  1  0  0
##           3  0  0 62  1  4
##           4  0  0  0 63  1
##           5  0  0  1  2 56
```

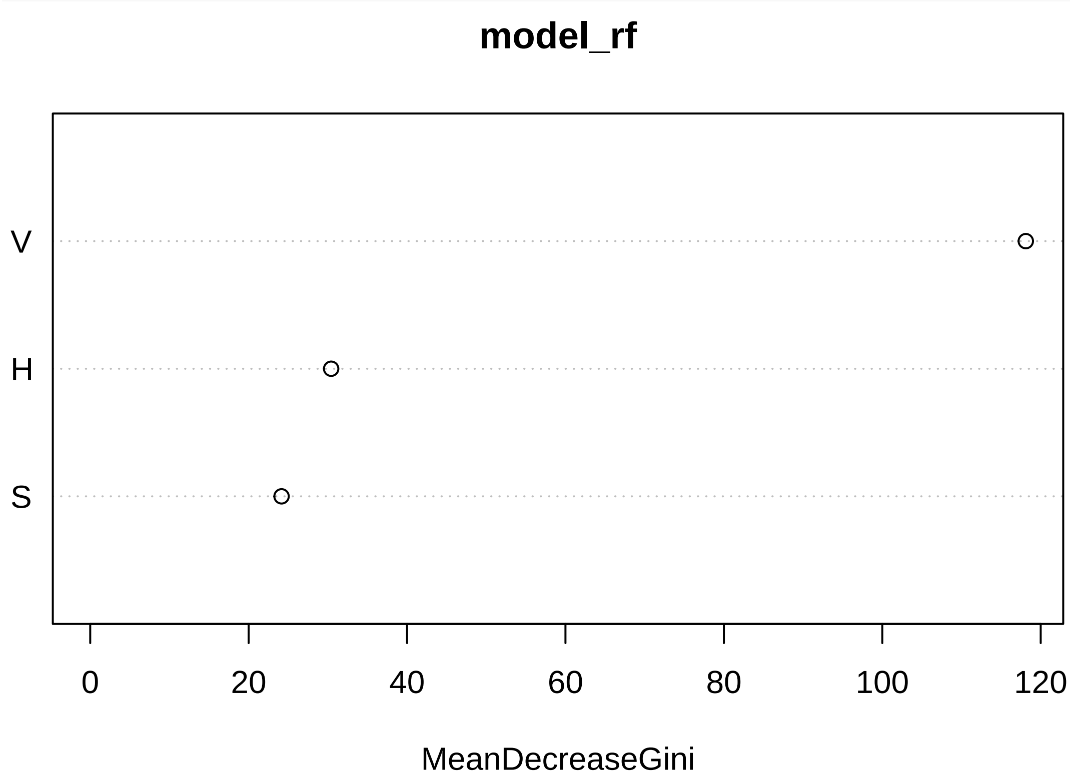
```r
accuracy <- sum(predictions == data$y) / nrow(data)
print(paste("Accuracy:", round(accuracy * 100, 2), "%"))
```

```
## [1] "Accuracy: 95.27 %"
```

```r
# Plotting variable importance
importance(model_rf)
```

```
##   MeanDecreaseGini
## V        118.11752
## H         30.41773
## S         24.15550
```

```r
varImpPlot(model_rf)
```



```r
# Confusion matrix for Random Forest model
pred <- predict(model_rf, newdata = X)
confusionMatrix(pred, y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5
##          1 71  0  3  0  4
##          2  0 70  1  0  0
##          3  0  0 61  1  4
##          4  0  0  0 63  1
##          5  0  0  1  2 56
##
## Overall Statistics
##
##                Accuracy : 0.9497
##                  95% CI : (0.9207, 0.9704)
##     No Information Rate : 0.2101
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9371
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            1.0000   1.0000   0.9242   0.9545   0.8615
## Specificity            0.9738   0.9963   0.9816   0.9963   0.9890
## Pos Pred Value         0.9103   0.9859   0.9242   0.9844   0.9492
## Neg Pred Value         1.0000   1.0000   0.9816   0.9891   0.9677
## Prevalence             0.2101   0.2071   0.1953   0.1953   0.1923
## Detection Rate         0.2101   0.2071   0.1805   0.1864   0.1657
## Detection Prevalence   0.2308   0.2101   0.1953   0.1893   0.1746
## Balanced Accuracy      0.9869   0.9981   0.9529   0.9754   0.9253
```

```r
train_control <- trainControl(method = "cv", number = 20)
model_rf_cv <- train(y ~ ., data = cbind(X, y), method = "rf", trControl = train_control)
```

```
## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
```

```r
print(model_rf_cv)
```

```
## Random Forest
##
## 338 samples
##   3 predictor
##   5 classes: '1', '2', '3', '4', '5'
##
## No pre-processing
## Resampling: Cross-Validated (20 fold)
## Summary of sample sizes: 320, 321, 322, 320, 322, 319, ...
## Resampling results across tuning parameters:
##
```

```
##   mtry  Accuracy   Kappa
##   2     0.5532652  0.4400585
##   3     0.5650127  0.4551959
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```