# GVault QDMS Post Go-Live Survey: R Notebook

## Initialize the packages

Remove comments if you haven't installed the packages in R yet.

```
# install.packages("RcURL")
# install.packages("randomForest")
# install.packages("e1071")
# install.packages("caret")
# install.packages("ggplot2")

library(RCurl)

## Loading required package: bitops

library(randomForest)

## randomForest 4.6-14

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

library(e1071)
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin

library(ggplot2)
set.seed(123)
```

Load dataset

```
df1<-read.csv("Gvault_survey_raw.csv",header = T)
```

## Data exploration

Drop irrelevant columns

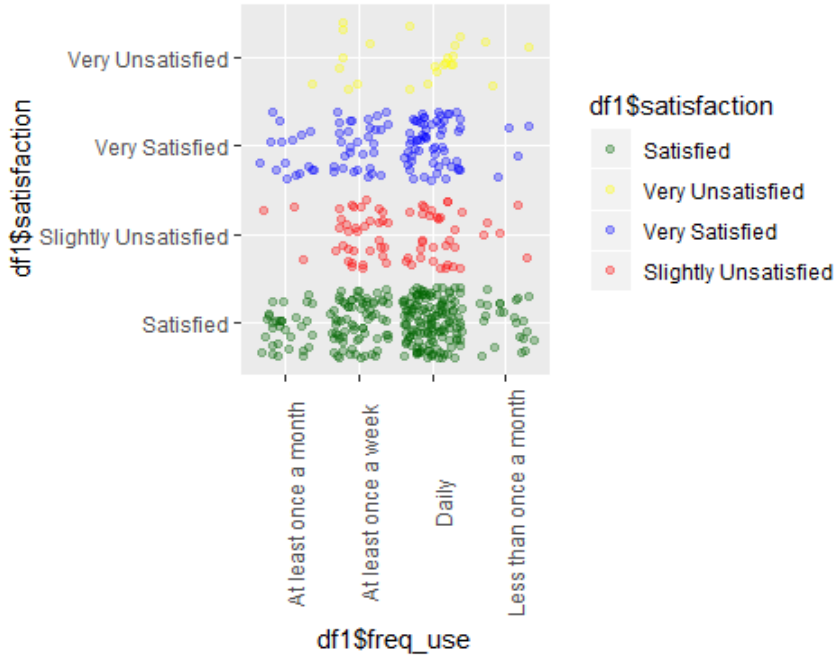
```
df1 = subset(df1, select = -c(Respondent.ID,Collector.ID,Start.Date,
          End.Date,explain_training_effectiveness,explain_support,
          explain_complete_without_help,explain_easy_access_documents,
          explain_satisfaction,explain_Gvault_efficiency,explain_Gvault_improved,
explain_office_location,explain_job_level) )
head(df1)

##                freq_use                 role training_instructor_led
## 1                 Daily       Owner / Author          Instructor Led
## 2  At least once a week              Not Sure
```

```
## 3                 Daily  Reviewer / Approver          Instructor Led
## 4 At least once a month  Reviewer / Approver          Instructor Led
## 5                 Daily Consumer - Read Only
## 6                 Daily  Reviewer / Approver          Instructor Led
##   training_web_based                                training_read no_training
## 1 Web/Computer Based Read & Understood of Procedural Document(s)
## 2 Web/Computer Based Read & Understood of Procedural Document(s)
## 3 Web/Computer Based Read & Understood of Procedural Document(s)
## 4
## 5                    Read & Understood of Procedural Document(s)
## 6 Web/Computer Based Read & Understood of Procedural Document(s)
##   training_effectiveness support_Gnet   support_inapplication
## 1              Effective
## 2   Not effective enough
## 3              Effective              In-Application (GVault)
## 4              Effective
## 5   Not effective enough
## 6              Effective         GNet
##                                                          support_ref_doc
## 1 Reference Document (User Manual, Reference Guide, Training Material, etc)
## 2 Reference Document (User Manual, Reference Guide, Training Material, etc)
## 3 Reference Document (User Manual, Reference Guide, Training Material, etc)
## 4
## 5
## 6 Reference Document (User Manual, Reference Guide, Training Material, etc)
##                  support_SOP                                 support_contacted
## 1 SOPs and Work Instructions Contacted my Document Control or Training Group
## 2
## 3
## 4 SOPs and Work Instructions
## 5
## 6 SOPs and Work Instructions Contacted my Document Control or Training Group
##   support_IT complete_without_help easy_access_documents   satisfaction
## 1                 Most of the time                   Yes      Satisfied
## 2                 Some of the time                   Yes      Satisfied
## 3                 Most of the time                   Yes      Satisfied
## 4                 Most of the time                   Yes Very Satisfied
## 5                     All the time                   Yes Very Satisfied
## 6                 Most of the time                   Yes      Satisfied
##          Gvault_efficiency Gvault_improved
## 1                Increased             Yes
## 2                Increased             Yes
## 3                Increased             Yes
## 4 No noticeable difference             Yes
## 5 No noticeable difference
## 6                Increased             Yes
##                                    functional_area explain_functional_area
## 1 Pharmaceutical Development and Manufacturing (PDM)
## 2                      Research and Development (R&D)
## 3 Pharmaceutical Development and Manufacturing (PDM)
## 4                           Facilities and Operations
## 5                      Research and Development (R&D)
## 6 Pharmaceutical Development and Manufacturing (PDM)
##   office_location              job_level time_worked_Glead
## 1     Foster City Manager / Group Leader          >7 Years
```

```
## 2       Cambridge                Director          >7 Years
## 3     Foster City Individual Contributor  Less than a year
## 4     Foster City Individual Contributor      1 to 2 years
## 5     Foster City Individual Contributor  Less than a year
## 6         Alberta Manager / Group Leader          >7 Years
```

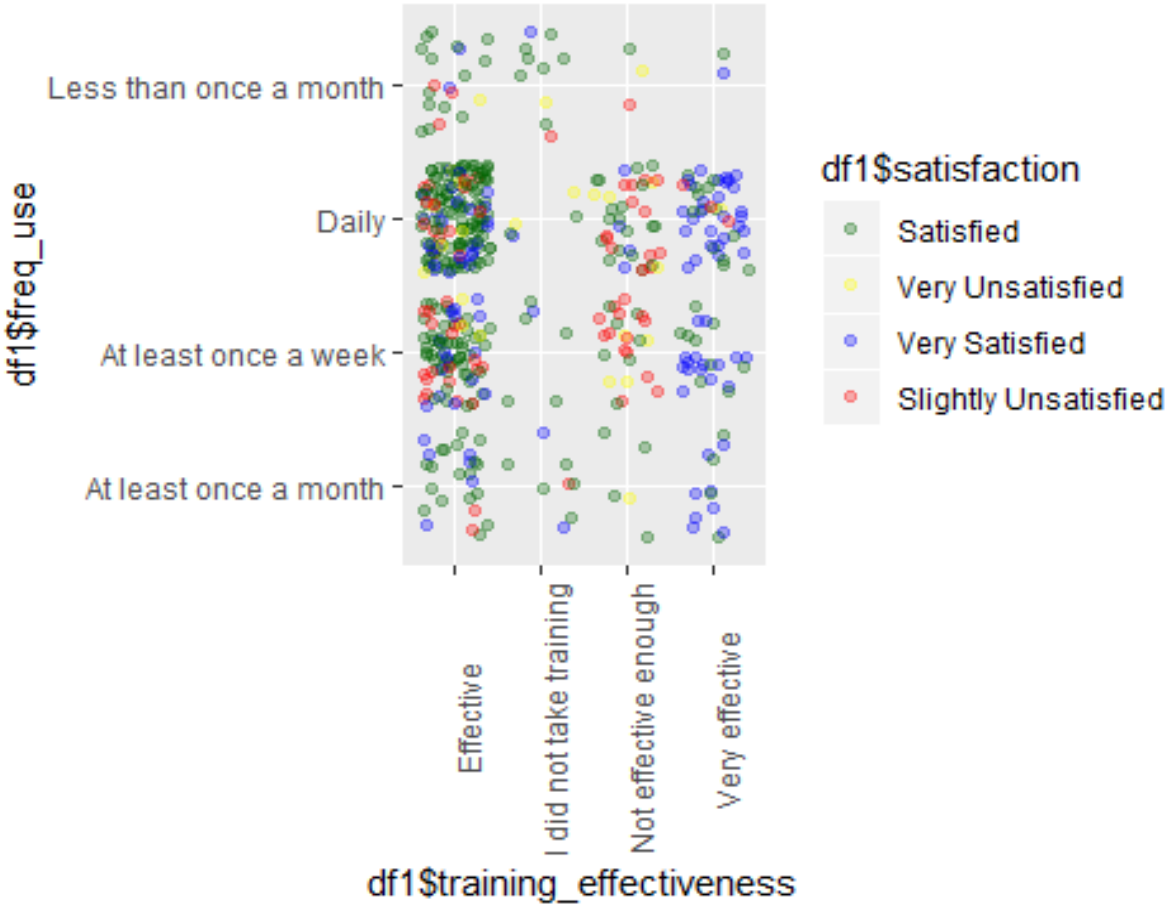
Explore the data before fitting a model to get an idea of what to expect. I am plotting a variable on two axes and using colors to see the relationship among the levels of satisfaction. Lets explore the relationship between satisfaction and frequency of use

```
p=ggplot(df1,aes(x=df1$freq_use,y=df1$satisfaction,
                 color=df1$satisfaction),width=120,height=60)+
  theme(axis.text.x = element_text(angle = 90))

p + geom_jitter(alpha=0.3) +
  scale_color_manual(breaks = c('Satisfied','Very Unsatisfied','Very Satisfied','Slightly
Unsatisfied'),
                     values=c('darkgreen','red','blue','yellow'))
```



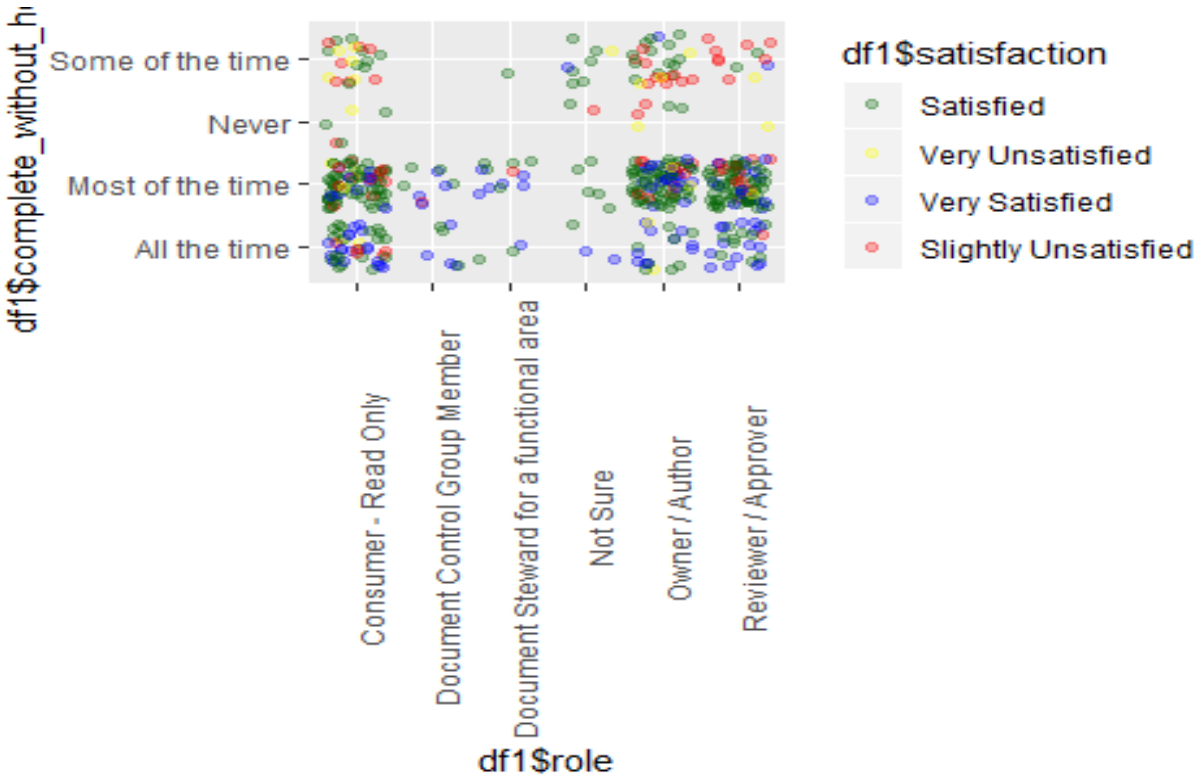

A comparison of Frequency of use (freq_use) & effectiveness of training to satisfaction shows us: Users who used Gvault daily were more likely to be satisfied or very satisfied. I'm looking for spots where there exists an overwhelming majority of one color.

```
p1=ggplot(df1,aes(y=df1$freq_use,x=df1$training_effectiveness,color=df1$satisfaction),wid
th=120,height=60)+theme(axis.text.x = element_text(angle = 90))
p1 + geom_jitter(alpha=0.3) +
  scale_color_manual(breaks = c('Satisfied','Very Unsatisfied','Very Satisfied','Slightly
Unsatisfied'),
                     values=c('darkgreen','red','blue','yellow'))
```




A comparison of role and the rate of completing work without help in terms likelihood of satisfaction show that: Reveiwer/Approver role users completed work in Gvault without help all the time and were more likely to be satisfied

```
p1=ggplot(df1,aes(x=df1$role,y=df1$complete_without_help,
color=df1$satisfaction),width=120,height=60)+theme(axis.text.x = element_text(angle =
90))
p1 + geom_jitter(alpha=0.3) +
  scale_color_manual(breaks = c('Satisfied','Very Unsatisfied','Very Satisfied','Slightly
Unsatisfied'),
                     values=c('darkgreen','red','blue','yellow'))
```




### Train test split

Create data for training

```
sample.ind = sample(2,nrow(df1),replace = T,prob = c(0.9,0.1))
data.dev = df1[sample.ind==1,]
data.val = df1[sample.ind==2,]
```

I wanted to know the split of satisfaction levels in the data set and compare it between the training and test data.

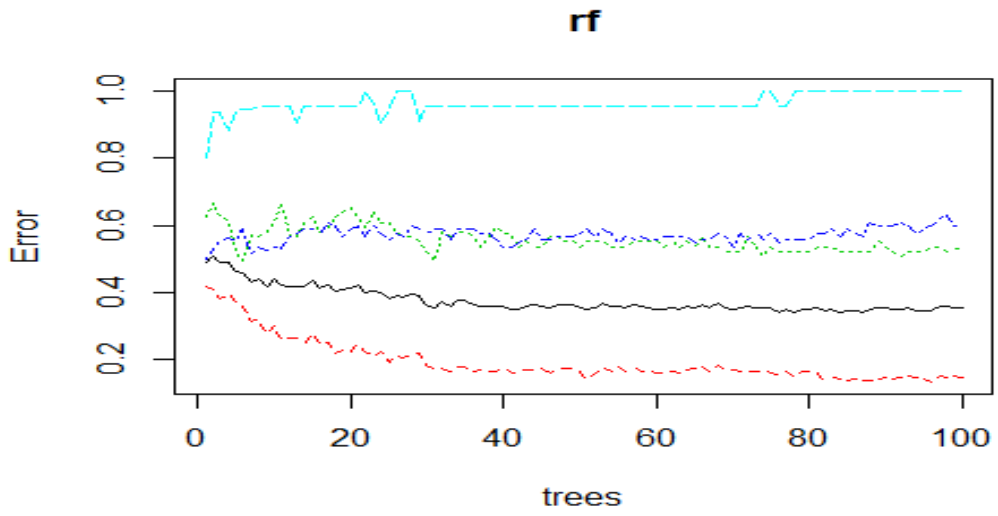
```
# Original Data
table(df1$satisfaction)/nrow(df1)

##
##            Satisfied Slightly Unsatisfied       Very Satisfied
##           0.56842105           0.14947368           0.23157895
##     Very Unsatisfied
##           0.05052632

# Training Data
table(data.dev$satisfaction)/nrow(data.dev)

##
##            Satisfied Slightly Unsatisfied       Very Satisfied
##           0.56206089           0.16159251           0.22716628
##     Very Unsatisfied
##           0.04918033

# Testing Data
table(data.val$satisfaction)/nrow(data.val)

##
##            Satisfied Slightly Unsatisfied       Very Satisfied
```

```
##            0.62500000           0.04166667           0.27083333
##     Very Unsatisfied
##            0.06250000
```

### Model Training: Fit Random Forest Model

I finally fit the random forest model to the training data. Plotting the model shows us that after about 20 trees, not much changes in terms of error. It fluctuates a bit but not to a large degree.

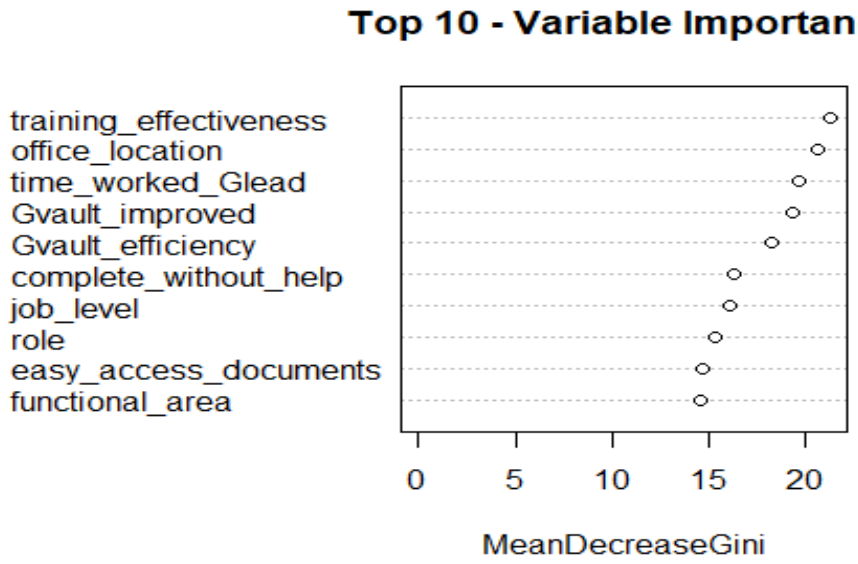
```
rf = randomForest(satisfaction ~ .,ntree = 100,data = data.dev)
plot(rf)
```




### Feature selection: Variable Importance

Training effectiveness is the most important variable in terms of "Mean Decreasing Gini" – a similar term for information gain.

```
varImpPlot(rf, sort = T, n.var=10,main="Top 10 - Variable Importance")
```



```r
var.imp = data.frame(importance(rf, type=2))
# make row names as columns
var.imp$Variables = row.names(var.imp)
print(var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),])
```

```
##                          MeanDecreaseGini               Variables
## training_effectiveness         21.285491  training_effectiveness
## office_location                20.574519         office_location
## time_worked_Glead              19.654643       time_worked_Glead
## Gvault_improved                19.349842         Gvault_improved
## Gvault_efficiency              18.195220       Gvault_efficiency
## complete_without_help          16.239664   complete_without_help
## job_level                      16.090225               job_level
## role                           15.298479                    role
## easy_access_documents          14.628286   easy_access_documents
## functional_area                14.518986         functional_area
## freq_use                       12.622863                freq_use
## support_ref_doc                 6.735509         support_ref_doc
## support_inapplication           6.655658   support_inapplication
## support_Gnet                    6.396340            support_Gnet
## support_SOP                     6.385338             support_SOP
## support_contacted               6.127248       support_contacted
## training_instructor_led         5.917755 training_instructor_led
## training_read                   5.645961           training_read
## training_web_based              5.619668      training_web_based
## explain_functional_area         3.812994 explain_functional_area
## support_IT                      3.025611              support_IT
## no_training                     1.363020             no_training
```

### Prediction and Model Evaluation

I decided to use the model to attempt to predict the satisfaction level based off of the training data set. It predicted the response variable perfectly – having zero false positives or false negatives.

```r
# Predicting response variable
data.dev$predicted.response = predict(rf , data.dev)

# Create Confusion Matrix
print(confusionMatrix(data = data.dev$predicted.response,
                      reference = data.dev$satisfaction,
                      positive ='Very Satisfied'))
```

```
## Confusion Matrix and Statistics
##
##                          Reference
## Prediction                Satisfied Slightly Unsatisfied Very Satisfied
##   Satisfied                     240                    0              2
##   Slightly Unsatisfied            0                   69              0
##   Very Satisfied                  0                    0             95
##   Very Unsatisfied                0                    0              0
##                          Reference
## Prediction                Very Unsatisfied
##   Satisfied                              0
##   Slightly Unsatisfied                   0
##   Very Satisfied                         0
##   Very Unsatisfied                      21
```

```
##
## Overall Statistics
##
##                Accuracy : 0.9953
##                  95% CI : (0.9832, 0.9994)
##     No Information Rate : 0.5621
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9922
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Satisfied Class: Slightly Unsatisfied
## Sensitivity                    1.0000                       1.0000
## Specificity                    0.9893                       1.0000
## Pos Pred Value                 0.9917                       1.0000
## Neg Pred Value                 1.0000                       1.0000
## Prevalence                     0.5621                       0.1616
## Detection Rate                 0.5621                       0.1616
## Detection Prevalence           0.5667                       0.1616
## Balanced Accuracy              0.9947                       1.0000
##                      Class: Very Satisfied Class: Very Unsatisfied
## Sensitivity                         0.9794                 1.00000
## Specificity                         1.0000                 1.00000
## Pos Pred Value                      1.0000                 1.00000
## Neg Pred Value                      0.9940                 1.00000
## Prevalence                          0.2272                 0.04918
## Detection Rate                      0.2225                 0.04918
## Detection Prevalence                0.2225                 0.04918
## Balanced Accuracy                   0.9897                 1.00000
```

### Model Testing

Now it was time to see how the model did with data it had not seen before– making predictions on the test data.

```
# Predicting response variable
data.val$predicted.response <- predict(rf ,data.val)

# Create Confusion Matrix
print(confusionMatrix(data=data.val$predicted.response,
                      reference=data.val$satisfaction,
                      positive='Very Satisfied'))
```

```
## Confusion Matrix and Statistics
##
##                          Reference
## Prediction                Satisfied Slightly Unsatisfied Very Satisfied
##   Satisfied                      26                     1              8
##   Slightly Unsatisfied            3                     1              0
##   Very Satisfied                  1                     0              5
##   Very Unsatisfied                0                     0              0
##                          Reference
## Prediction                Very Unsatisfied
```

```
##   Satisfied                           0
##   Slightly Unsatisfied                3
##   Very Satisfied                      0
##   Very Unsatisfied                    0
##
## Overall Statistics
##
##                Accuracy : 0.6667
##                  95% CI : (0.5159, 0.796)
##     No Information Rate : 0.625
##     P-Value [Acc > NIR] : 0.3313
##
##                   Kappa : 0.3391
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Satisfied Class: Slightly Unsatisfied
## Sensitivity                    0.8667                      0.50000
## Specificity                    0.5000                      0.86957
## Pos Pred Value                 0.7429                      0.14286
## Neg Pred Value                 0.6923                      0.97561
## Prevalence                     0.6250                      0.04167
## Detection Rate                 0.5417                      0.02083
## Detection Prevalence           0.7292                      0.14583
## Balanced Accuracy              0.6833                      0.68478
##                      Class: Very Satisfied Class: Very Unsatisfied
## Sensitivity                         0.3846                   0.0000
## Specificity                         0.9714                   1.0000
## Pos Pred Value                      0.8333                      NaN
## Neg Pred Value                      0.8095                   0.9375
## Prevalence                          0.2708                   0.0625
## Detection Rate                      0.1042                   0.0000
## Detection Prevalence                0.1250                   0.0000
## Balanced Accuracy                   0.6780                   0.5000
```