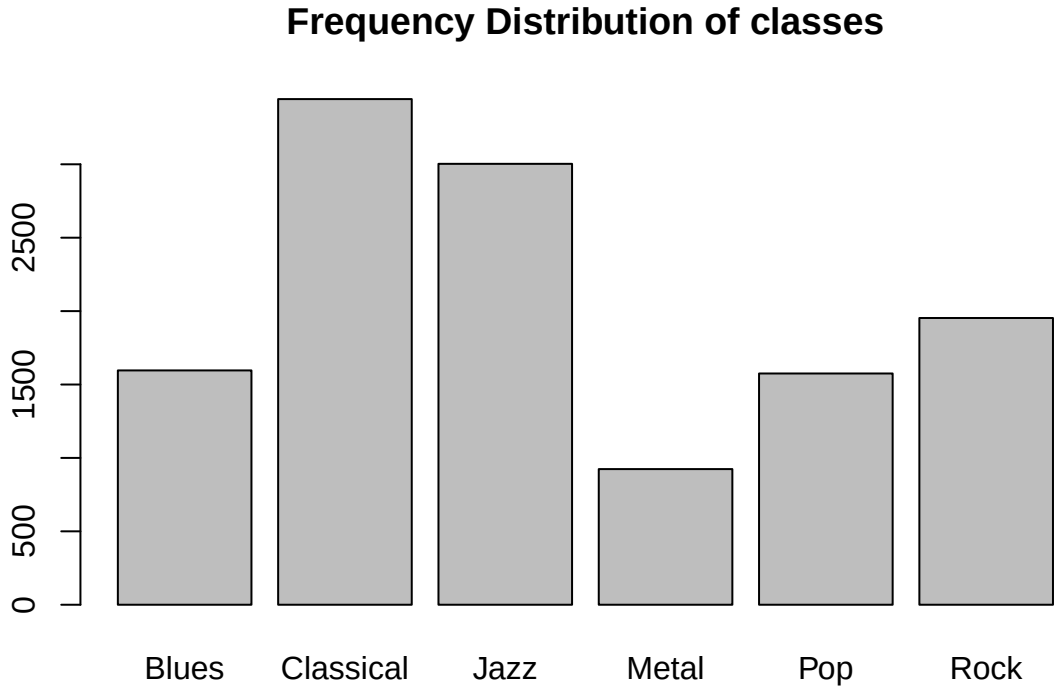
Applied Predictive Modeling Ex4

Chathura J Gunasekara

Monday, September 29, 2014

4.1 (a) In this situation, since the sample size is large it is possible to set aside a testing data set and training data set. simple random spliting , Stratified random sampling or maximum dissimilarity sampling should be used because the dispropotionality of the classes.

```
## Loading required package: lattice
## Loading required package: ggplot2
```




```
##
## Attaching package: 'proxy'
##
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

(b)Code for implementing : stratified sampling

Using “createDataPartition” function in Caret package.

```
##      Resample1
## [1,]         2
## [2,]         7
## [3,]        14
## [4,]        20
## [5,]        22
## [6,]        47
```

implementting maximum dissimilarity sampling in caret package. The data will be split on the basis of the predictor values.

```
## A random sample of 5 data points
startSet <- sample(1:dim(train_music)[1], 5)
samplePool <- train_music[-startSet, ]
start <- train_music[startSet, ]
newSamp <- maxDissim(start, samplePool, n = 4)
```

4.2. (a) This data set is small and can not find a frequency distribution because there is no classes in this data set.So it not possible to do stratified sampling or random sampling.Bootstrapping, k - fold cross validation , Repeated cross validation is used.

```
##  num [1:165, 1:1108] 12.52 1.12 19.41 1.73 1.68 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:165] "1" "2" "3" "4" ...
##   ..$ : chr [1:1108] "permeability" "X1" "X2" "X3" ...
```

(b)

Code for implementation :Repeated stratified sampling

```
library(caret)
set.seed(1)
repeatSplits <-createDataPartition(permeability,p=0.8,times=3)
str(repeatSplits)
```

```
## List of 3
##  $ Resample1: int [1:133] 2 6 8 13 17 25 29 48 77 87 ...
##  $ Resample2: int [1:133] 2 6 8 11 13 17 29 45 48 49 ...
##  $ Resample3: int [1:133] 2 6 8 11 13 17 25 29 45 48 ...
```

Code for implementation :10 fold cross validation.

```
set.seed(1)
cvSplits <-createFolds(permeability,k=10,returnTrain=TRUE)

str(cvSplits)
```

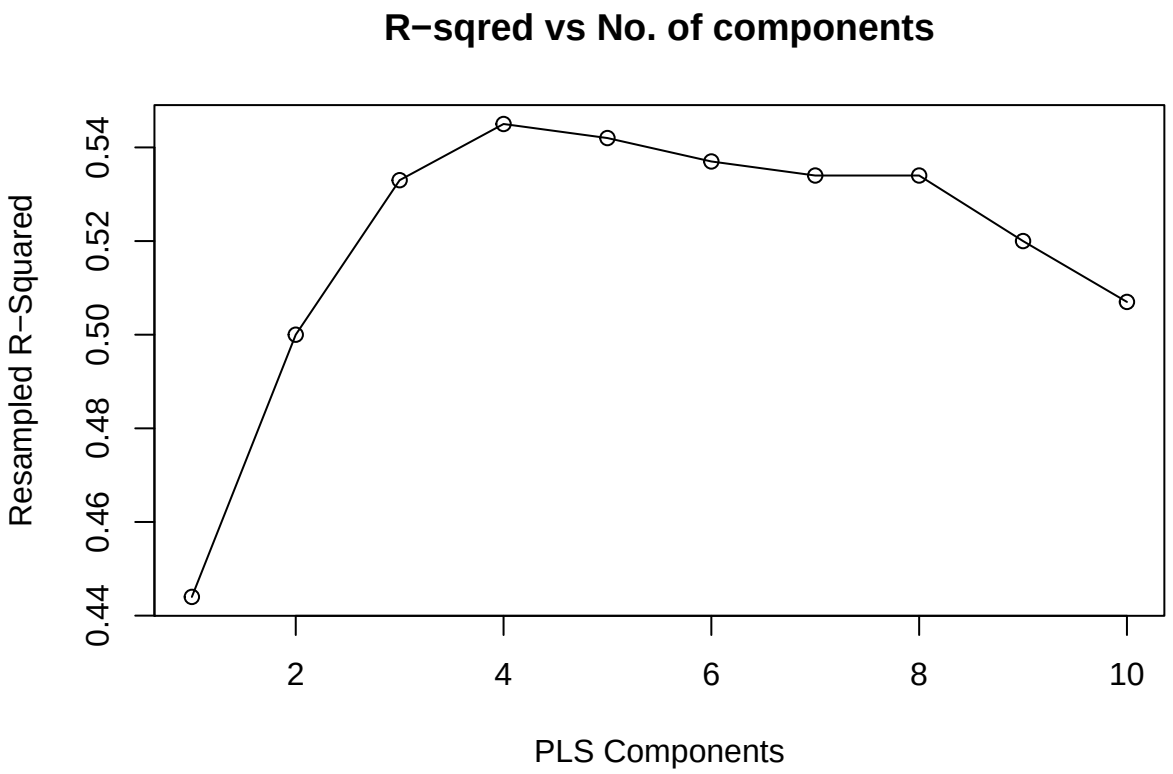
```
## List of 10
##  $ Fold01: int [1:149] 1 2 3 4 5 6 7 8 10 11 ...
##  $ Fold02: int [1:148] 1 2 3 4 5 6 8 9 10 11 ...
##  $ Fold03: int [1:148] 1 2 4 6 7 9 10 11 12 13 ...
##  $ Fold04: int [1:149] 1 2 3 4 5 6 7 8 9 10 ...
```

```
##  $ Fold05: int [1:149] 1 2 3 4 5 7 8 9 10 11 ...
##  $ Fold06: int [1:148] 1 2 3 5 6 7 8 9 11 12 ...
##  $ Fold07: int [1:149] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Fold08: int [1:147] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Fold09: int [1:149] 2 3 4 5 6 7 8 9 10 11 ...
##  $ Fold10: int [1:149] 1 3 4 5 6 7 8 9 10 11 ...
```

4.3. This data set contains information about a chemical manufacturing process, in which the goal is to understand the relationship between the process and the resulting final product yield. Raw material in this process is put through a sequence of 27 steps to make the final pharmaceutical product. The starting material is generated from a biological unit and has a range of quality and characteristics. The objective in this project was to develop a model to predict percent yield of the manufacturing process. The data set consisted of 177 samples of biological material for which 57 characteristics were measured.

(a)A parsimonious model is a model that accomplishes a desired level of explanation or prediction with as few predictor variables as possible.

Follwing plot shows the R-squared and number of PLS components in the model.



Models with components 1 to 4 increase R squared. then after 4 it decreases becasue over fitting. Numercically optimal value is 0.545 its SD is 0.0308 onestandar_error = 0.545-0.0308

```
0.545-0.0308
```

```
## [1] 0.5142
```

0.533 > 0.5142 , which is within one standard deviation so Number of PLS Components is enough to model is 3.

b)Computing the toerance values.

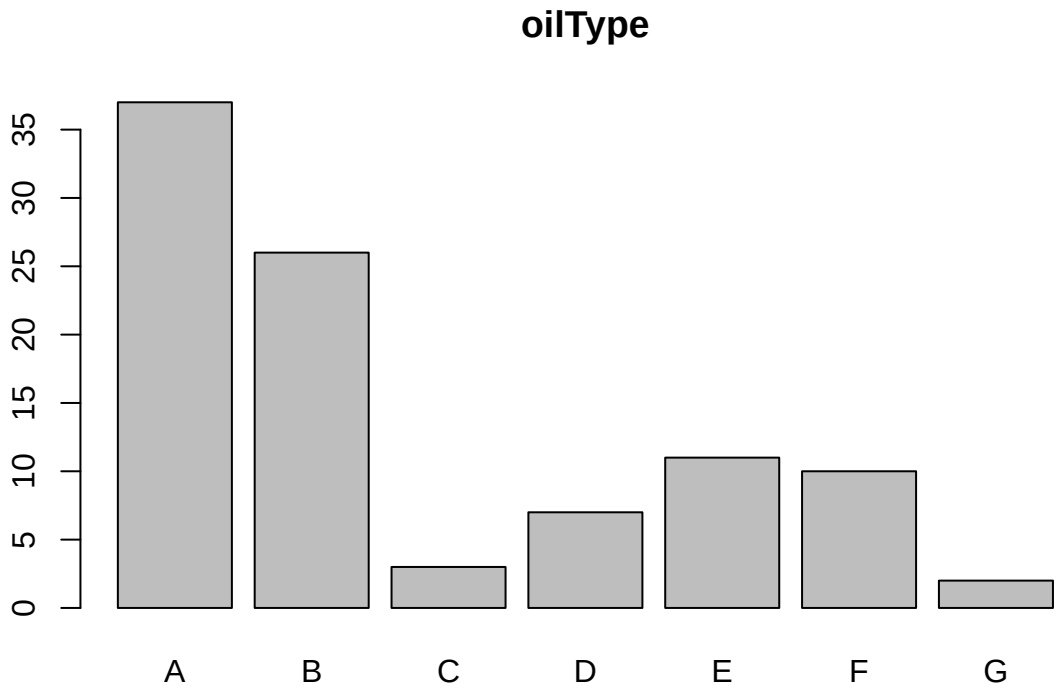
Optimal number of PLS components if 10% loss in R-squared acceptable is 2

c) SVM : R Squared is higher in SVM and Random Forests Not much defferece in them, but predictin time is way high for Random forests.so in concering R squared SVM is better. d)

consider using the simplest model that reasonably approximates the performance of more complex methods with a acceptable prediction time. So in that perspective KNN is better.

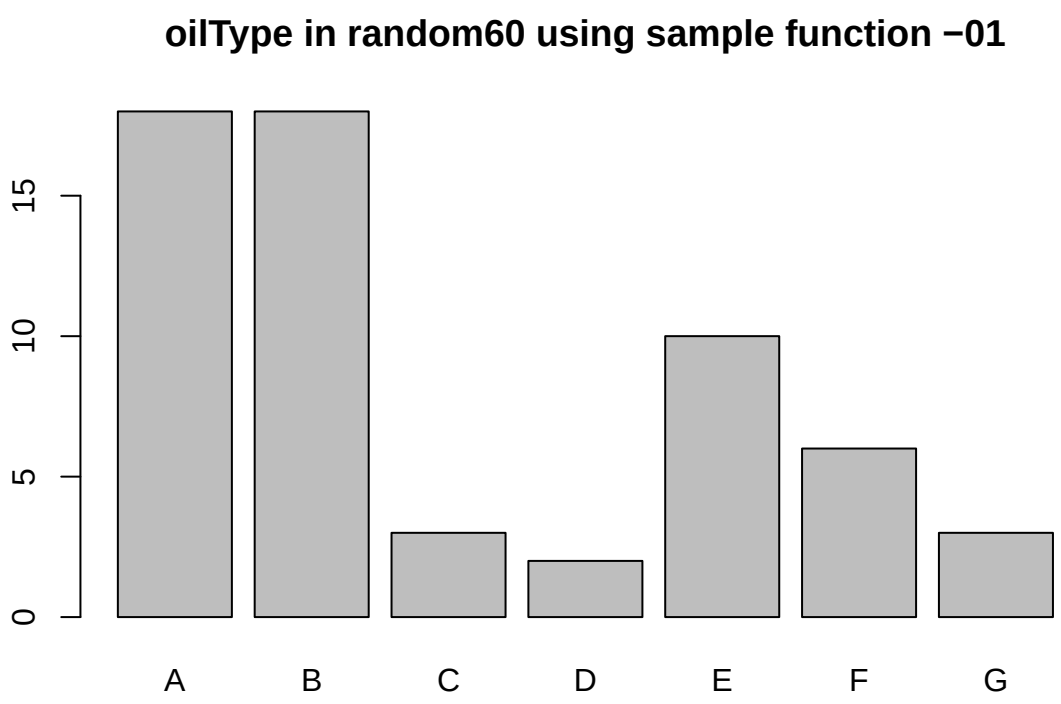
4.4

```
##  Factor w/ 7 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

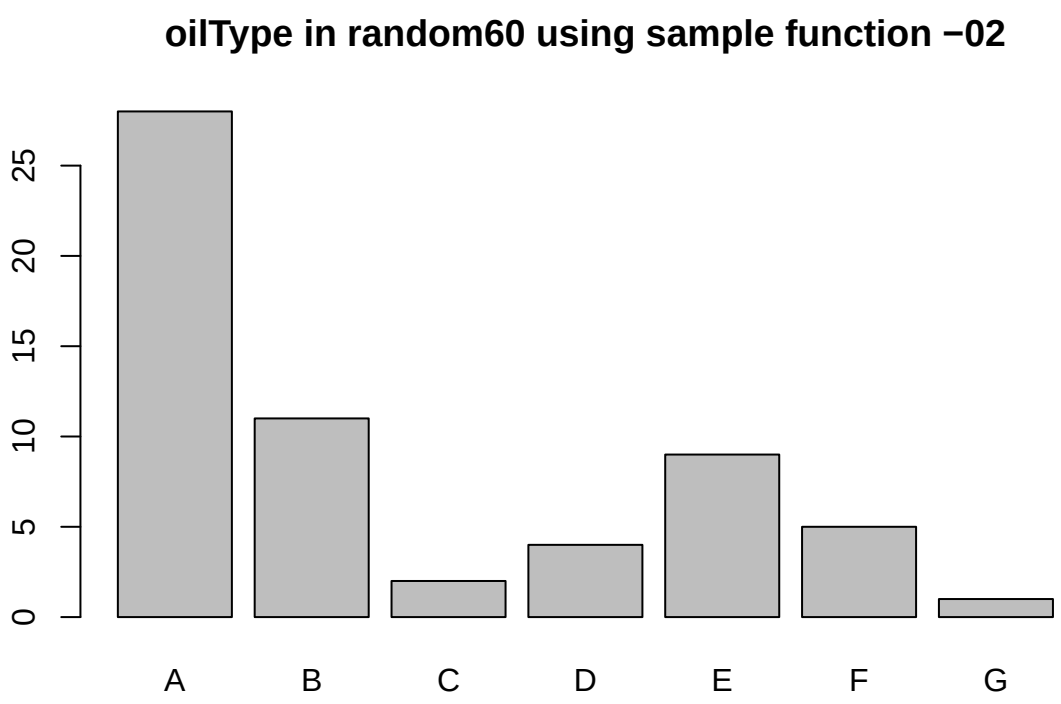


(a)Using Sample function to create a random sample of 60 oils.

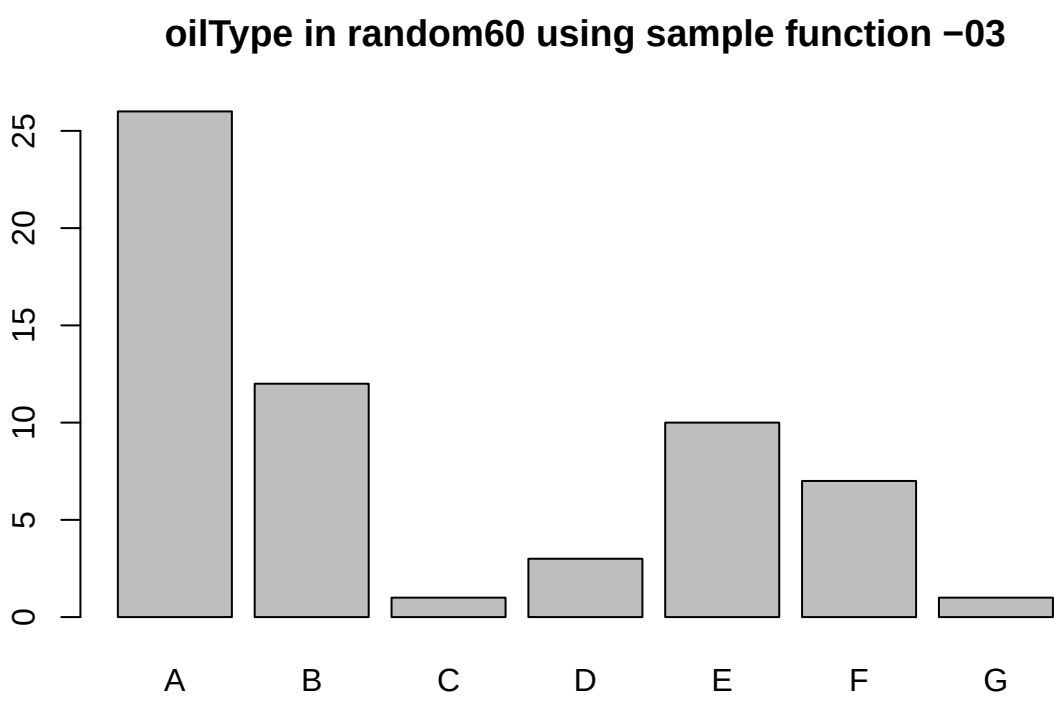
```
barplot(table(oilType[sample((1:96),60,replace=TRUE,prob=NULL )]),main="oilType in random60 using sampl
```



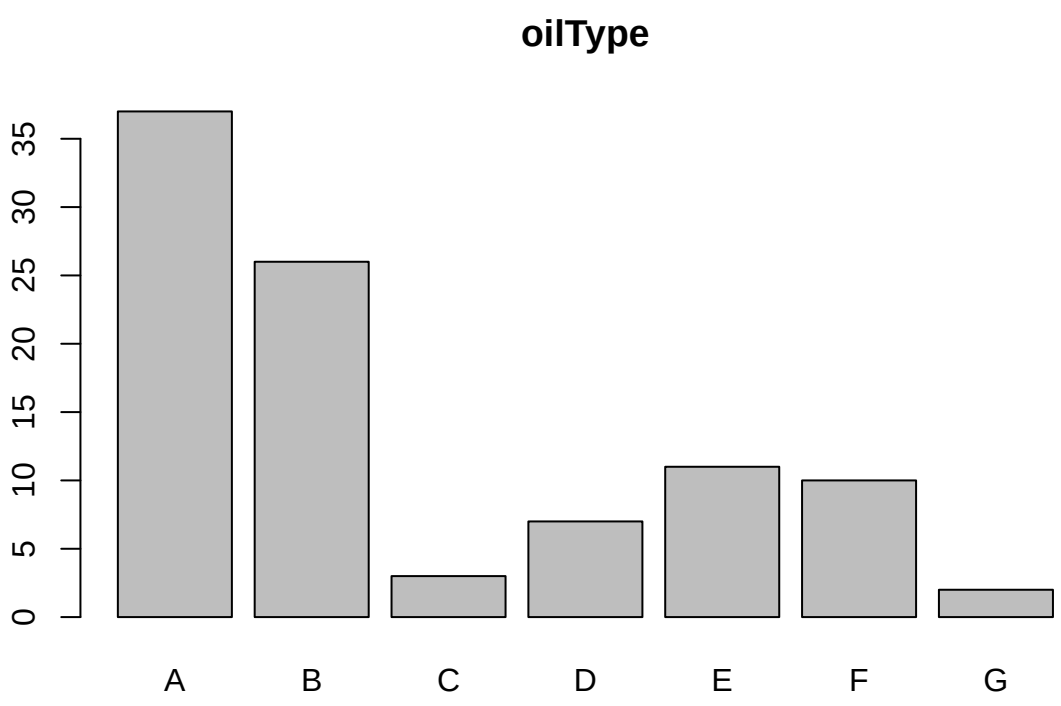

```
barplot(table(oilType[sample((1:96),60,replace=TRUE,prob=NULL )]),main="oilType in random60 using sampl
```

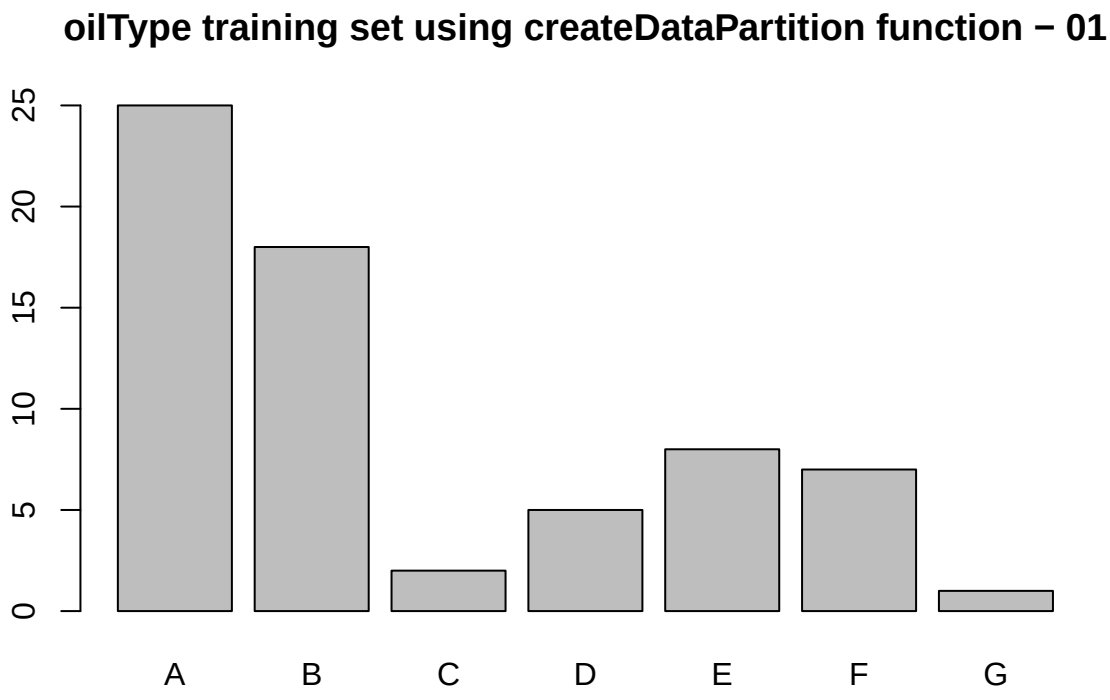


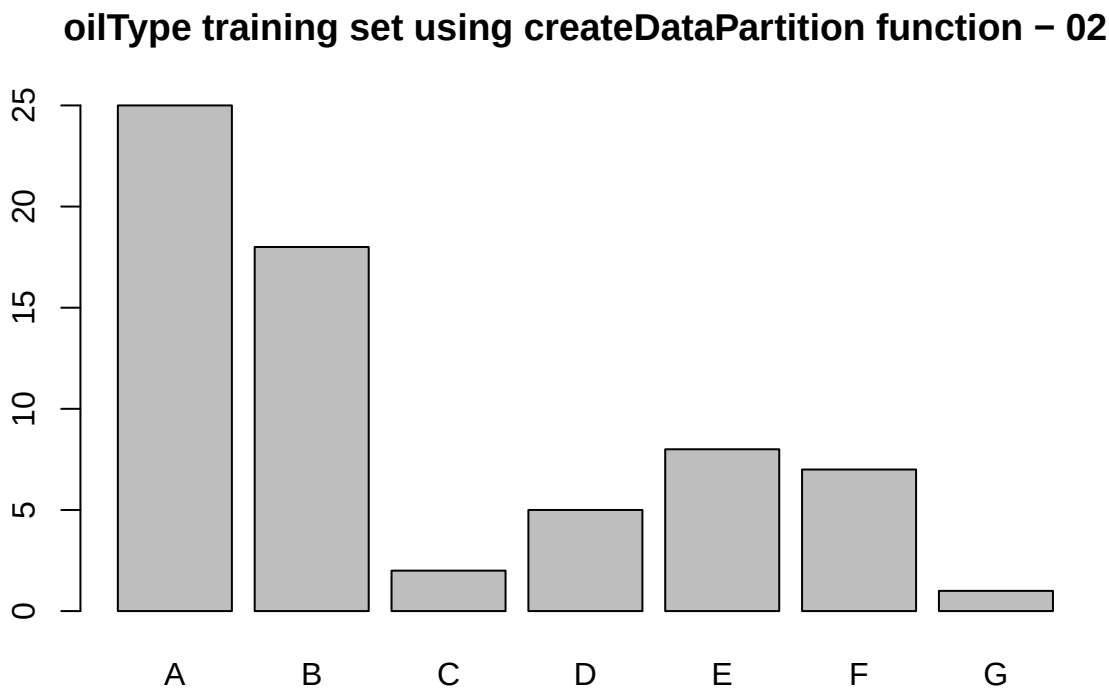

```
barplot(table(oilType[sample((1:96),60,replace=TRUE,prob=NULL )]),main="oilType in random60 using sampl
```



Base on the above figures it is observable that the variation in the random sampling. Sometimes very few obeservations of a class can be selected and sometimes an entire class may be not selected.When one class has a disproportionately small frequency compared to the others, there is a chance that the distribution of the outcomes may be substantially different between the training and test sets. (b)

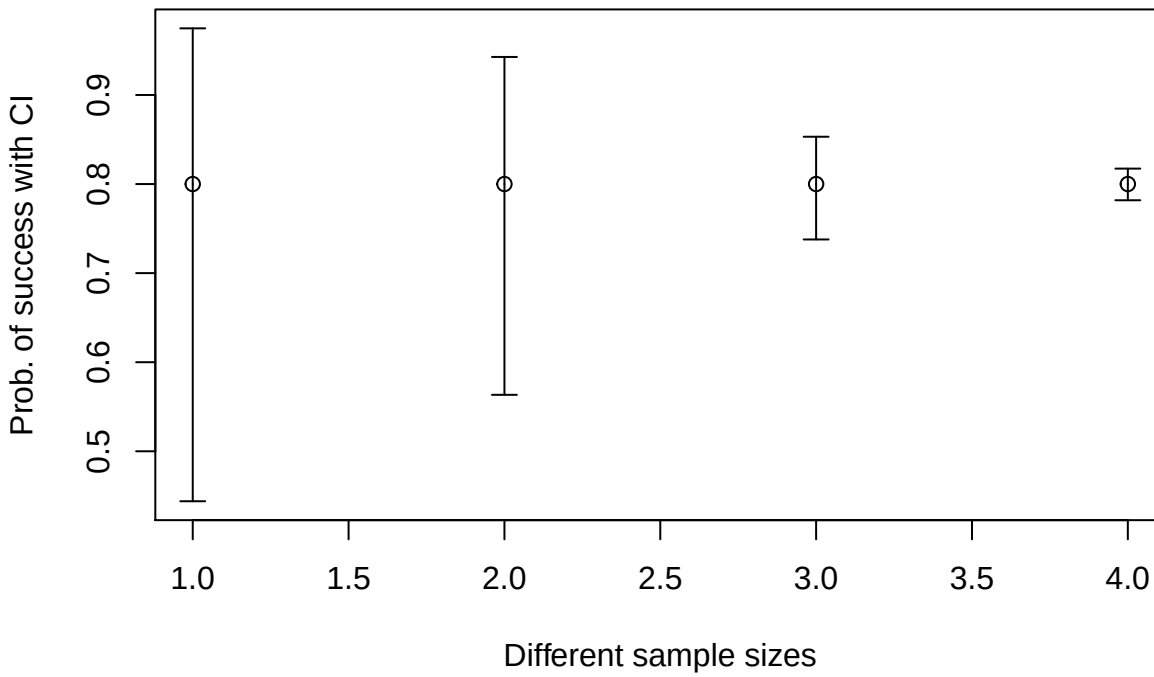

oilType
0
5
10
15
20
25
30
35
A
B
C
D
E
F
G


oilType training set using createDataPartition function − 01
0
5
10
15
20
25
A
B
C
D
E
F
G

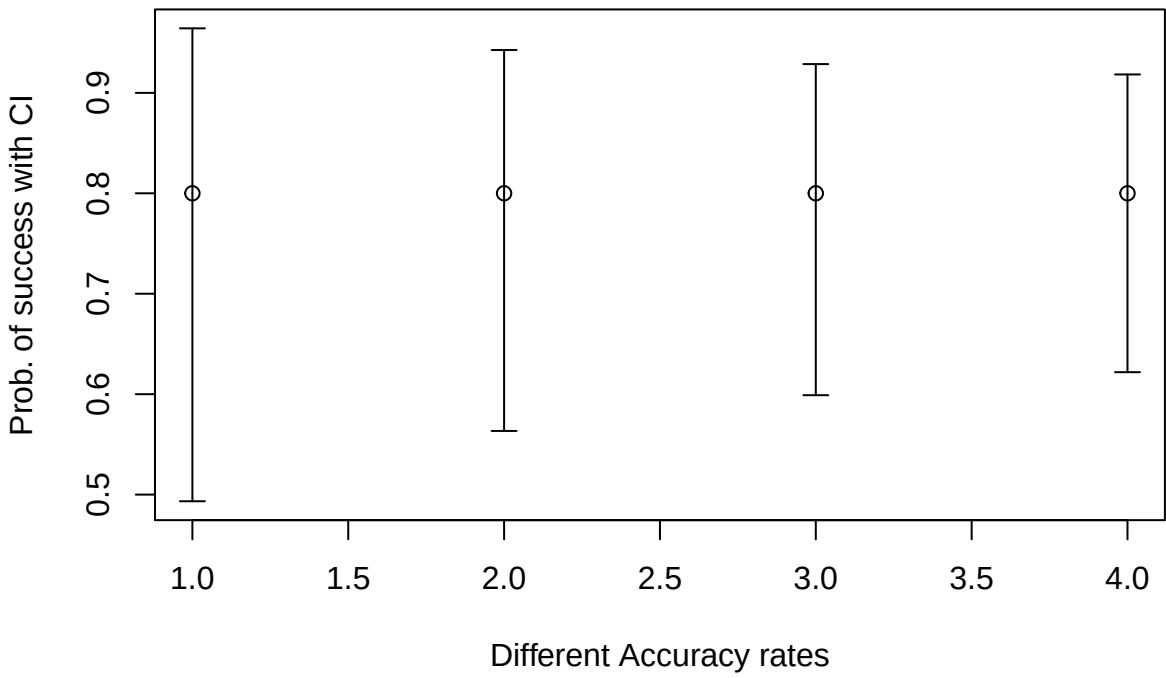


This graph does not change as random sample and it preserve the frequency distribution of the original sample In this way, there is a higher likelihood that the outcome distributions will match.

(c) A test set should be avoided because the sample size is small , random test set may not be enough give sufficient power or precision to make a judgement. So validation using a single test set can be a poor choise. Best to use resampling methods such as Cross validation which evaluates many alternate versions of the data. k-Fold Cross-Validation Repeated Training/Test Splits The Bootstrap



(d) Different sample sizes
When test set size increase uncertainity decrease.



Different accuracy rates
when accuracy rate is decreases is doesn't make a huge difference in uncertainity.