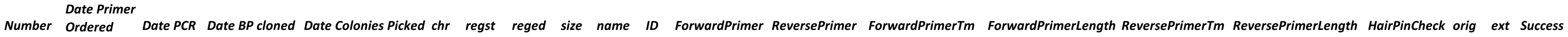
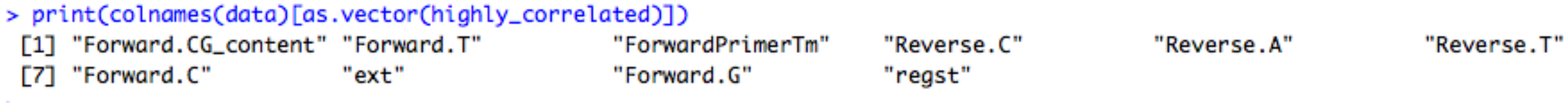
- Raw data [20 Features | 236 Data Points | 0-1 Success Labels]



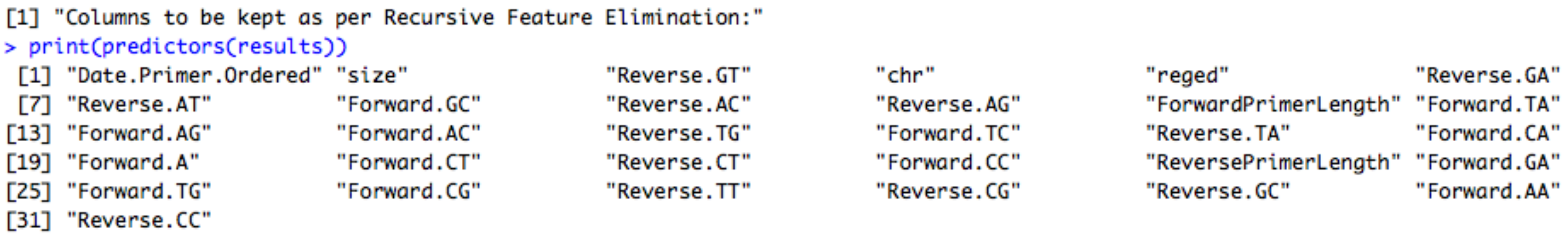
Number	*Date Primer Ordered*	*Date PCR*	*Date BP cloned*	*Date Colonies Picked*	*chr*	*regst*	*reged*	*size*	*name*	*ID*	*ForwardPrimer*	*ReversePrimer*	*ForwardPrimerTm*	*ForwardPrimerLength*	*ReversePrimerTm*	*ReversePrimerLength*	*HairPinCheck*	*orig*	*ext*	*Success*

- Preprocessing
  - Remove ID and un-useful columns (*Number*, *ID*, etc.)
  - Add forward & reverse counts for bases and all possible *k*-mers with *k* = 2 counts (+(8 + 2x16) = 40 columns)
  - Add forward & reverse CG content (+2 columns)
  - Total number of columns = 52
- Feature Selection
  - High Correlation | 10 columns discarded



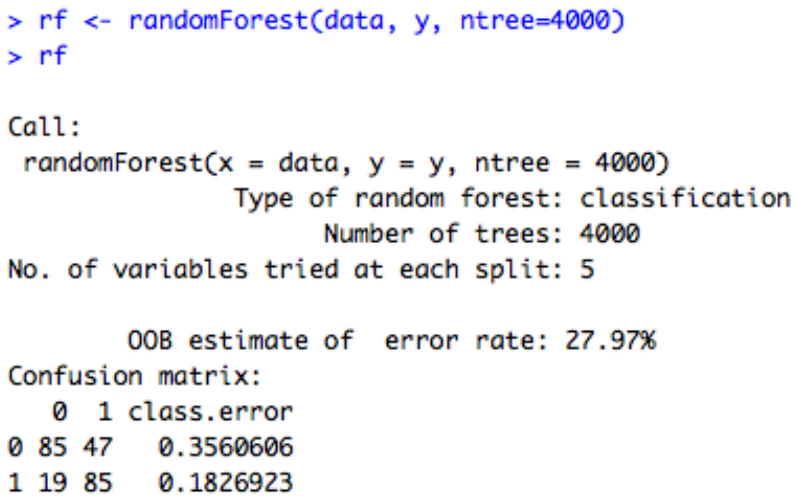
```
> print(colnames(data)[as.vector(highly_correlated)])
 [1] "Forward.CG_content" "Forward.T"          "ForwardPrimerTm"    "Reverse.C"          "Reverse.A"          "Reverse.T"
 [7] "Forward.C"          "ext"                "Forward.G"          "regst"
```

  - Recursive Feature Elimination | 32 significant columns kept



```
[1] "Columns to be kept as per Recursive Feature Elimination:"
> print(predictors(results))
 [1] "Date.Primer.Ordered" "size"                "Reverse.GT"          "chr"                 "reged"               "Reverse.GA"
 [7] "Reverse.AT"          "Forward.GC"          "Reverse.AC"          "Reverse.AG"          "ForwardPrimerLength" "Forward.TA"
[13] "Forward.AG"          "Forward.AC"          "Reverse.TG"          "Forward.TC"          "Reverse.TA"          "Forward.CA"
[19] "Forward.A"           "Forward.CT"          "Reverse.CT"          "Forward.CC"          "ReversePrimerLength" "Forward.GA"
[25] "Forward.TG"          "Forward.CG"          "Reverse.TT"          "Reverse.CG"          "Reverse.GC"          "Forward.AA"
[31] "Reverse.CC"
```

- Random Forest
  - 100-5000 trees tested | 4000 trees performed best
  - *Performance*
    - *Precision* 0.8173077
    - *Accuracy* 0.720339
    - +/- 0.03 as dataset is small



```
> rf <- randomForest(data, y, ntree=4000)
> rf

Call:
 randomForest(x = data, y = y, ntree = 4000)
               Type of random forest: classification
                     Number of trees: 4000
No. of variables tried at each split: 5

        OOB estimate of  error rate: 27.97%
Confusion matrix:
   0  1 class.error
0 85 47   0.3560606
1 19 85   0.1826923
```

- Room for further improvement | Suggestions for features?
- Forward & reverse counts of *k*-mers with *k* = 3 didn’t help
- R script can score new data points | Easy to run by Mark R’s lab members