Variable Selection

Mladen Kolar (mkolar@chicagobooth.edu)

Variable Selection

For many models, predictive performance is degraded as the number of uninformative predictors increases.

Simpler model requires less computational resources.

Model is more interpretable with fewer predictors.

Classes of variable selection techniques

Intrinsic methods

Examples: tree based models, regularization based methods, such as the lasso

Pros: no external feature selection tool is required

Cons: These approaches are specific to a model being used.

Filter methods

Evaluate the relevance of predictors using some statistic (e.g., information gain, odds-ratio, χ^2 statistics, correlation). Only keep predictors that pass some threshold criterion.

Typically, each feature is viewed as independent of the others, effectively ignoring interactions between features.

Pros: computationally efficient

Cons: redundant predictors may be selected; hard to detect interaction and nonlinear effects; a selection of predictors that meets a filtering may not be a set that improves predictive performance.

Wrapper methods

Deterministic wrapper feature selection methods either start with no features or with all features included in the model and iteratively refine the set of chosen features according to some model quality measures.

- forward-selection; backward-selection (*recursive feature elimination* or RFE)

Stochastic wrapper feature selection procedures

- genetic algorithms (GA); simulated annealing (SA).

Wrappers have the potential advantage of searching a wider variety of predictor subsets than filters.

- computationally demanding—need to fit a potentially time consuming model multiple times.

Summary: The different types of feature selection methods have their own pros and cons.

- It is important to remember that the globally best subset is often difficult to find.

The Effect of Irrelevant Features

The effect depends on:

- the type of model;
- the nature of the predictors;
- the ratio of the size of the training set to the number of predictors.

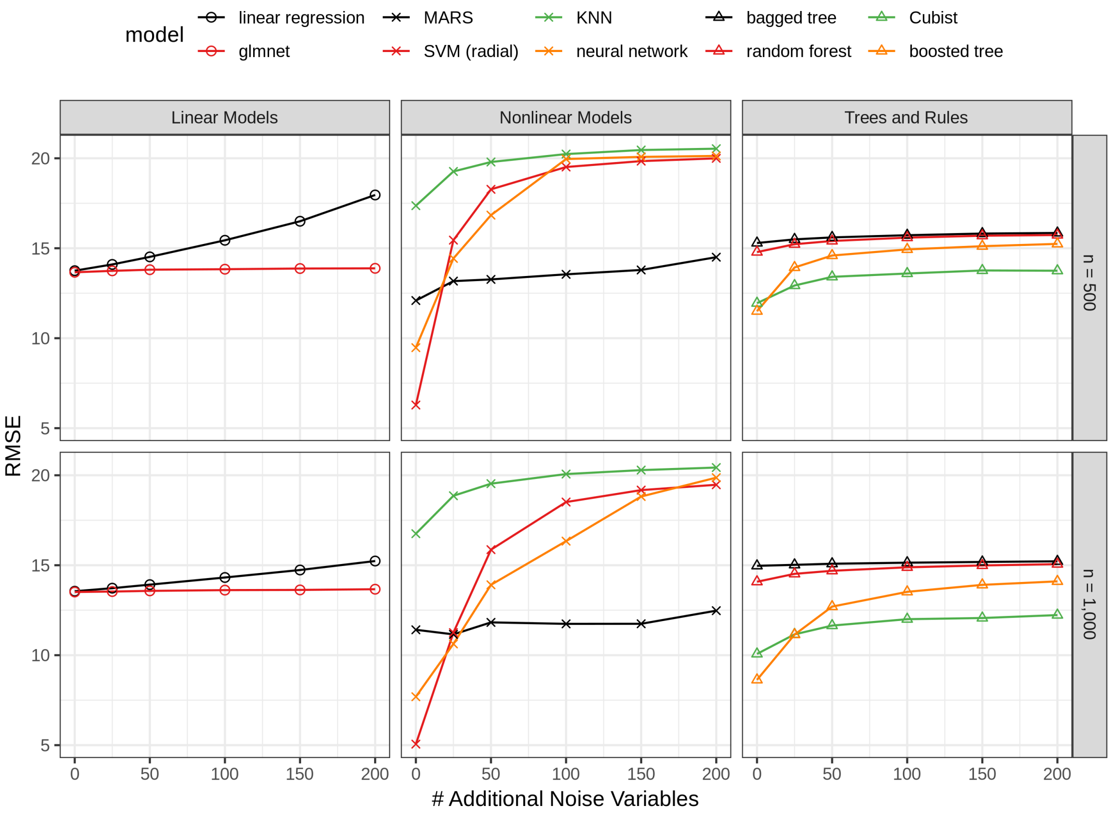
Let us take a look at a simulation in Chapter 10.3 of **Feature Engineering and Selection** by *Max Kuhn and Kjell Johnson* (https://bookdown.org/max/FES/feature-selection-simulation.html).

- The code for the simulation is here: https://github.com/topepo/FES_Selection_Simulation
- The simulation setting is taken from Sapp, Laan, and Canny (2014) .

$$\begin{aligned} y =& x_1 + \sin(x_2) + \log(|x_3|) + x_4^2 + x_5 x_6 + I(x_7 x_8 x_9 < 0) + I(x_{10} > 0) + \\ & x_{11} I(x_1 1 > 0) + \sqrt(|x_{12}|) + \cos(x_{13}) + 2x_{14} + |x_{15}| + I(x_{16} < -1) + \\ & x_{17} I(x_{17} < -1) - 2x_{18} - x_{19}x_{20} + \epsilon \end{aligned} \tag{1}$$

- input variables $x_j \sim N(0, 1)$; the error $\epsilon \sim N(0, 3^2)$;
- between 10 and 200 extra columns of input variables with no connection to the outcome were added;
- the training size $n = 500$ or $n = 1,000$.

The focus of simulation is to look at the relative impact of irrelevant predictors and not at absolute performance.



Overfitting When Selecting Input Variables

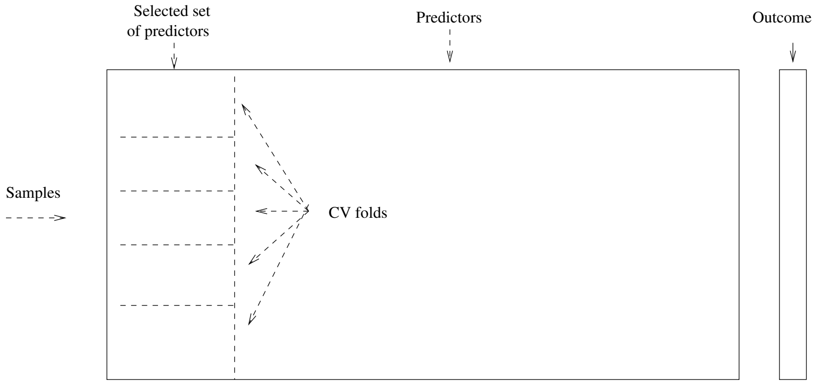
Often it is possible to find a subset of input variables that has good predictive performance on the training set

- but has *poor performance* when used on a **test set**.

Feature selection needs to be part of the cross-validation (or more generally resampling) **process.**

Wrong approach

The **most common mistake** is to only conduct cross-validation inside of the variable selection procedure.



Algorithm:

```
1. Rank the predictors using the training set;
2. For each subset size, $S_i$
   2.1 For each split into training/validation set
       • Fit model with $S_i$ most important variables on the training set.
       • Predict the validation set.
   2.2 Calculate the model performance with $S_i$ variables
3. Determine the appropriate number of predictors (i.e., the $S_i$ with best performance)
4. Fit the final model based on the optimal $S_i$
```

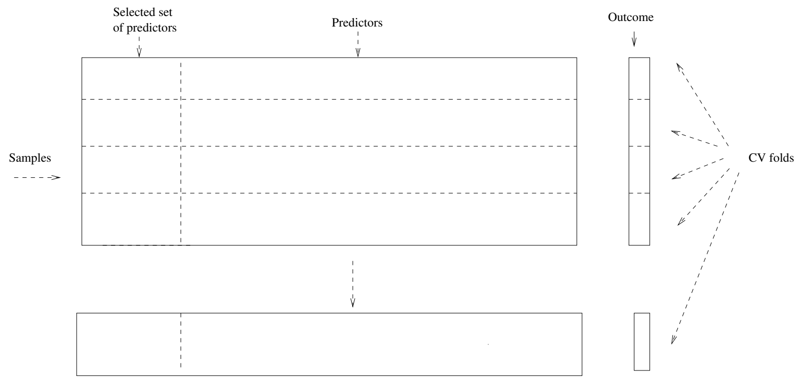
Two key problems with the above procedure:

- The feature selection is performed outside cross-validation. CV cannot effectively measure the impact of the selection process.
- The same data are being used to measure performance of the model and to select the input variables This is the same issue that arises when fitting a model to the training set and then using the same training set to measure performance. There is an obvious bias in measuring the performance of the model if it can closely fit the training data. We need out-of-sample data to accurately determine how well the model is doing. If the variable selection process results in overfitting, there are no data remaining that could possibly inform us of the problem.

Correct approach

We must include variable selection as a component of the modeling process.

- In the same way that we are choosing other tuning parameters for the model.



Algorithm:

1. For each fold of cross-validation, $1 \ldots k$
    1. Split data into training/validation set
    2. Rank the predictors using the training set
    3. **For** *each subset size*, S_i
        - Fit model with S_i most important variables on the training set.
        - Predict the validation set.
2. Calculate the model performance with S_i variables
3. Determine the appropriate number of variables
4. Fit the final model based on the optimal number of variables using the original training set

In the above procedure, the number of variables to select is treated as a tuning parameter.

- feature selection is done within cross-validation;
- different set of input variables may be selected on each one of the cross-validation folds;
- computational cost increases.

Large data sets tend to greatly reduce the risk of overfitting to the predictors during variable selection. Using separate data splits for variable ranking/filtering, modeling, and evaluation can be both efficient and effective.

Example: Amyotrophic Lateral Sclerosis

We are going to explore the following two variable selection procedures

- `Boruta` package: stochastic wrapper procedure that uses random forest to compute variable importance measures
- Recursive Feature Elimination: a classical deterministic wrapper method

The data is from this paper:

- Model-Based and Model-Free Techniques for Amyotrophic Lateral Sclerosis Diagnostic Prediction and Patient Clustering. by Tang, M., Gao, C, Goutman, SA, Kalinin, A, Mukherjee, B, Guan, Y, and Dinov, ID.

Amyotrophic Lateral Sclerosis (ALS) is a rare but devastating disease. The data are from a large clinical trial including big, multi-source and heterogeneous datasets. The clinical data shows that the rate of ALS progression varies significantly among patients. Majority of the patients die within 3 to 5 years after ALS onset, however, a few are able survive for over 10 years. This heterogeneity of disease course hinders demonstration of its biological mechanism and development of effective treatment.

We need to develop reliable predictive models of ALS progression to understand the pathophysiology of the disease.

The dataset contains 2,223 observations and 131 numeric variables. We select `ALSFRS_slope` as our outcome variable, as it captures the patients' clinical decline over a year.

```
ALS.train<-read.csv("ALS_TrainingData_2223.csv")
summary(ALS.train)
```

```
##        ID          Age_mean     Albumin_max   Albumin_median  Albumin_min   Albumin_range
##  Min.   :   1   Min.   :18.0   Min.   :37.0   Min.   :34.5   Min.   :24.0   Min.   :0.0000
##  1st Qu.: 614   1st Qu.:47.0   1st Qu.:45.0   1st Qu.:42.0   1st Qu.:39.0   1st Qu.:0.0090
##  Median :1213   Median :55.0   Median :47.0   Median :44.0   Median :41.0   Median :0.0121
##  Mean   :1215   Mean   :54.6   Mean   :47.0   Mean   :44.0   Mean   :40.8   Mean   :0.0138
##  3rd Qu.:1816   3rd Qu.:63.0   3rd Qu.:49.0   3rd Qu.:46.0   3rd Qu.:43.0   3rd Qu.:0.0159
##  Max.   :2424   Max.   :81.0   Max.   :70.3   Max.   :51.1   Max.   :49.0   Max.   :0.2439
##   ALSFRS_slope   ALSFRS_Total_max ALSFRS_Total_median ALSFRS_Total_min ALSFRS_Total_range
##  Min.   :-4.35   Min.   :11.0     Min.   : 2.5        Min.   : 0.0     Min.   :0.0000
##  1st Qu.:-1.09   1st Qu.:29.0     1st Qu.:23.0        1st Qu.:14.0     1st Qu.:0.0140
##  Median :-0.62   Median :33.0     Median :28.0        Median :20.0     Median :0.0233
##  Mean   :-0.73   Mean   :31.7     Mean   :27.1        Mean   :19.9     Mean   :0.0260
##  3rd Qu.:-0.28   3rd Qu.:36.0     3rd Qu.:32.0        3rd Qu.:27.0     3rd Qu.:0.0348
##  Max.   : 1.21   Max.   :40.0     Max.   :40.0        Max.   :40.0     Max.   :0.1176
##  ALT.SGPT._max ALT.SGPT._median ALT.SGPT._min   ALT.SGPT._range AST.SGOT._max AST.SGOT._median
##  Min.   : 10   Min.   :  8      Min.   :  1.6   Min.   :0.003   Min.   : 11   Min.   :  9.0
##  1st Qu.: 32   1st Qu.: 22      1st Qu.: 15.0   1st Qu.:0.030   1st Qu.: 30   1st Qu.: 22.0
##  Median : 45   Median : 30      Median : 21.0   Median :0.048   Median : 38   Median : 27.0
##  Mean   : 54   Mean   : 33      Mean   : 23.0   Mean   :0.071   Mean   : 43   Mean   : 29.1
##  3rd Qu.: 65   3rd Qu.: 40      3rd Qu.: 28.0   3rd Qu.:0.078   3rd Qu.: 48   3rd Qu.: 34.0
##  Max.   :944   Max.   :193      Max.   :109.0   Max.   :2.383   Max.   :911   Max.   :100.0
##  AST.SGOT._min  AST.SGOT._range Bicarbonate_max Bicarbonate_median Bicarbonate_min
##  Min.   : 1.0   Min.   :0.000   Min.   :20.0    Min.   :19.5       Min.   : 2.5
##  1st Qu.:17.0   1st Qu.:0.024   1st Qu.:29.0    1st Qu.:26.0       1st Qu.:22.0
##  Median :20.0   Median :0.035   Median :31.0    Median :27.0       Median :23.0
##  Mean   :21.5   Mean   :0.049   Mean   :30.9    Mean   :27.0       Mean   :23.2
##  3rd Qu.:25.0   3rd Qu.:0.052   3rd Qu.:32.0    3rd Qu.:28.0       3rd Qu.:24.4
##  Max.   :86.0   Max.   :1.917   Max.   :52.0    Max.   :39.5       Max.   :34.0
##  Bicarbonate_range Blood.Urea.Nitrogen..BUN._max Blood.Urea.Nitrogen..BUN._median
##  Min.   :0.0000    Min.   : 2.92                 Min.   : 2.19
##  1st Qu.:0.0127    1st Qu.: 5.84                 1st Qu.: 4.64
##  Median :0.0149    Median : 6.94                 Median : 5.42
##  Mean   :0.0169    Mean   : 7.35                 Mean   : 5.56
##  3rd Qu.:0.0181    3rd Qu.: 8.21                 3rd Qu.: 6.35
##  Max.   :0.2143    Max.   :25.19                 Max.   :11.87
##  Blood.Urea.Nitrogen..BUN._min Blood.Urea.Nitrogen..BUN._range bp_diastolic_max bp_diastolic_median
##  Min.   : 0.58                 Min.   :0.0000                  Min.   : 70      Min.   : 56.0
##  1st Qu.: 3.29                 1st Qu.:0.0041                  1st Qu.: 88      1st Qu.: 78.0
##  Median : 4.07                 Median :0.0058                  Median : 90      Median : 80.0
##  Mean   : 4.16                 Mean   :0.0071                  Mean   : 92      Mean   : 81.1
##  3rd Qu.: 5.00                 3rd Qu.:0.0084                  3rd Qu.: 98      3rd Qu.: 85.0
##  Max.   :10.22                 Max.   :0.0695                  Max.   :140      Max.   :110.0
##  bp_diastolic_min bp_diastolic_range bp_systolic_max bp_systolic_median bp_systolic_min
##  Min.   : 20.0    Min.   :0.000      Min.   :100     Min.   : 90        Min.   : 72
##  1st Qu.: 65.0    1st Qu.:0.035      1st Qu.:138     1st Qu.:120        1st Qu.:108
##  Median : 70.0    Median :0.043      Median :145     Median :130        Median :110
##  Mean   : 69.9    Mean   :0.048      Mean   :147     Mean   :130        Mean   :113
##  3rd Qu.: 75.0    3rd Qu.:0.054      3rd Qu.:157     3rd Qu.:136        3rd Qu.:120
##  Max.   :100.0    Max.   :0.714      Max.   :220     Max.   :190        Max.   :165
##  bp_systolic_range  Calcium_max   Calcium_median  Calcium_min    Calcium_range      Chloride_max
##  Min.   :0.000     Min.   :2.17   Min.   :2.05   Min.   :0.244   Min.   :0.00000   Min.   : 96
##  1st Qu.:0.053     1st Qu.:2.40   1st Qu.:2.28   1st Qu.:2.171   1st Qu.:0.00037   1st Qu.:106
##  Median :0.065     Median :2.47   Median :2.35   Median :2.230   Median :0.00047   Median :107
##  Mean   :0.071     Mean   :2.47   Mean   :2.35   Mean   :2.223   Mean   :0.00054   Mean   :107
##  3rd Qu.:0.082     3rd Qu.:2.53   3rd Qu.:2.40   3rd Qu.:2.298   3rd Qu.:0.00059   3rd Qu.:109
##  Max.   :0.405     Max.   :9.46   Max.   :2.80   Max.   :2.650   Max.   :0.01290   Max.   :119
##  Chloride_median  Chloride_min   Chloride_range   Creatinine_max  Creatinine_median Creatinine_min
##  Min.   : 90     Min.   : 76.0   Min.   :0.0000   Min.   : 22.0   Min.   : 18.0     Min.   :  0.0
##  1st Qu.:102     1st Qu.: 98.0   1st Qu.:0.0125   1st Qu.: 65.0   1st Qu.: 53.0     1st Qu.: 39.0
##  Median :104     Median :100.0   Median :0.0159   Median : 79.6   Median : 62.0     Median : 53.0
##  Mean   :104     Mean   : 99.3   Mean   :0.0179   Mean   : 78.8   Mean   : 65.2     Mean   : 52.0
##  3rd Qu.:105     3rd Qu.:101.0   3rd Qu.:0.0199   3rd Qu.: 88.4   3rd Qu.: 78.8     3rd Qu.: 61.9
##  Max.   :111     Max.   :109.0   Max.   :0.2143   Max.   :248.0   Max.   :176.8     Max.   :168.0
##  Creatinine_range  Gender_mean    Glucose_max   Glucose_median   Glucose_min    Glucose_range
##  Min.   :0.000    Min.   :1.00   Min.   : 4.2   Min.   : 3.50   Min.   : 0.00   Min.   :0.0000
##  1st Qu.:0.038    1st Qu.:1.00   1st Qu.: 5.8   1st Qu.: 4.91   1st Qu.: 4.05   1st Qu.:0.0031
##  Median :0.049    Median :2.00   Median : 6.5   Median : 5.30   Median : 4.44   Median :0.0047
##  Mean   :0.058    Mean   :1.64   Mean   : 7.2   Mean   : 5.49   Mean   : 4.27   Mean   :0.0063
##  3rd Qu.:0.070    3rd Qu.:2.00   3rd Qu.: 7.6   3rd Qu.: 5.70   3rd Qu.: 4.80   3rd Qu.:0.0074
##  Max.   :0.421    Max.   :2.00   Max.   :33.7   Max.   :26.20   Max.   :12.20   Max.   :0.0975
##    hands_max     hands_median    hands_min     hands_range     Hematocrit_max Hematocrit_median
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.0000   Min.   : 0.4   Min.   : 0.4
##  1st Qu.:5.00   1st Qu.:3.00   1st Qu.:0.00   1st Qu.:0.0036   1st Qu.:42.3   1st Qu.:40.0
##  Median :7.00   Median :5.50   Median :3.00   Median :0.0067   Median :45.2   Median :42.6
##  Mean   :6.18   Mean   :4.91   Mean   :3.05   Mean   :0.0069   Mean   :41.9   Mean   :39.5
##  3rd Qu.:8.00   3rd Qu.:7.00   3rd Qu.:5.00   3rd Qu.:0.0095   3rd Qu.:47.7   3rd Qu.:45.0
##  Max.   :8.00   Max.   :8.00   Max.   :8.00   Max.   :0.0429   Max.   :81.0   Max.   :56.0
##  Hematocrit_min Hematocrit_range Hemoglobin_max Hemoglobin_median Hemoglobin_min  Hemoglobin_range
##  Min.   : 0.3   Min.   :0.0000   Min.   :116    Min.   :106       Min.   :  6.2   Min.   :0.000
##  1st Qu.:37.0   1st Qu.:0.0072   1st Qu.:144    1st Qu.:136       1st Qu.:128.0   1st Qu.:0.023
##  Median :40.0   Median :0.0097   Median :152    Median :145       Median :136.0   Median :0.031
##  Mean   :37.0   Mean   :0.0114   Mean   :152    Mean   :144       Mean   :135.5   Mean   :0.038
##  3rd Qu.:42.7   3rd Qu.:0.0136   3rd Qu.:160    3rd Qu.:152       3rd Qu.:145.0   3rd Qu.:0.042
##  Max.   :52.9   Max.   :0.1857   Max.   :280    Max.   :182       Max.   :180.0   Max.   :0.562
##     leg_max       leg_median      leg_min       leg_range        mouth_max     mouth_median
##  Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.0000   Min.   : 1.0   Min.   : 0.0
##  1st Qu.:3.00   1st Qu.:2.50   1st Qu.:1.00   1st Qu.:0.0034   1st Qu.:10.0   1st Qu.: 8.0
##  Median :5.00   Median :3.00   Median :2.00   Median :0.0054   Median :12.0   Median :11.0
##  Mean   :5.31   Mean   :4.05   Mean   :2.49   Mean   :0.0062   Mean   :10.7   Mean   : 9.7
##  3rd Qu.:8.00   3rd Qu.:6.00   3rd Qu.:3.00   3rd Qu.:0.0087   3rd Qu.:12.0   3rd Qu.:12.0
##  Max.   :8.00   Max.   :8.00   Max.   :8.00   Max.   :0.0420   Max.   :12.0   Max.   :12.0
##    mouth_min      mouth_range     onset_delta_mean onset_site_mean Platelets_max Platelets_median
##  Min.   : 0.00   Min.   :0.0000   Min.   :-3119    Min.   :1.0     Min.   : 84   Min.   : 73
##  1st Qu.: 5.00   1st Qu.:0.0018   1st Qu.: -887    1st Qu.:2.0     1st Qu.:239   1st Qu.:204
##  Median : 9.00   Median :0.0053   Median : -572    Median :2.0     Median :275   Median :233
##  Mean   : 7.78   Mean   :0.0066   Mean   : -683    Mean   :1.8     Mean   :285   Mean   :239
##  3rd Qu.:11.00   3rd Qu.:0.0103   3rd Qu.: -374    3rd Qu.:2.0     3rd Qu.:320   3rd Qu.:270
##  Max.   :12.00   Max.   :0.0368   Max.   :  -16    Max.   :3.0     Max.   :866   Max.   :526
##  Platelets_min Potassium_max  Potassium_median Potassium_min  Potassium_range    pulse_max
##  Min.   :  0   Min.   : 3.4   Min.   :3.00     Min.   :2.40   Min.   :0.0000   Min.   : 53.0
##  1st Qu.:175   1st Qu.: 4.4   1st Qu.:4.00     1st Qu.:3.70   1st Qu.:0.0011   1st Qu.: 84.0
##  Median :204   Median : 4.5   Median :4.20     Median :3.90   Median :0.0014   Median : 90.0
##  Mean   :208   Mean   : 4.6   Mean   :4.19     Mean   :3.86   Mean   :0.0017   Mean   : 90.6
##  3rd Qu.:236   3rd Qu.: 4.8   3rd Qu.:4.30     3rd Qu.:4.00   3rd Qu.:0.0019   3rd Qu.: 96.0
##  Max.   :476   Max.   :43.0   Max.   :5.10     Max.   :5.10   Max.   :0.0987   Max.   :144.0
##   pulse_median   pulse_min      pulse_range    respiratory_max respiratory_median respiratory_min
##  Min.   : 50   Min.   : 18.0   Min.   :0.005   Min.   :2.00    Min.   :0.00       Min.   :0.00
##  1st Qu.: 72   1st Qu.: 60.0   1st Qu.:0.037   1st Qu.:4.00    1st Qu.:3.00       1st Qu.:2.00
##  Median : 77   Median : 64.0   Median :0.049   Median :4.00    Median :4.00       Median :3.00
##  Mean   : 77   Mean   : 65.4   Mean   :0.054   Mean   :3.91    Mean   :3.59       Mean   :2.79
##  3rd Qu.: 81   3rd Qu.: 70.0   3rd Qu.:0.062   3rd Qu.:4.00    3rd Qu.:4.00       3rd Qu.:4.00
##  Max.   :115   Max.   :102.0   Max.   :0.500   Max.   :4.00    Max.   :4.00       Max.   :4.00
##  respiratory_range   Sodium_max  Sodium_median   Sodium_min   Sodium_range      SubjectID
##  Min.   :0.00000   Min.   :134   Min.   :128   Min.   :112   Min.   :0.0000   Min.   :   533
##  1st Qu.:0.00000   1st Qu.:142   1st Qu.:139   1st Qu.:135   1st Qu.:0.0106   1st Qu.:240826
##  Median :0.00183   Median :143   Median :140   Median :137   Median :0.0131   Median :496835
##  Mean   :0.00251   Mean   :143   Mean   :140   Mean   :137   Mean   :0.0150   Mean   :498880
##  3rd Qu.:0.00365   3rd Qu.:145   3rd Qu.:141   3rd Qu.:138   3rd Qu.:0.0173   3rd Qu.:750300
##  Max.   :0.02542   Max.   :169   Max.   :146   Max.   :145   Max.   :0.1429   Max.   :999482
##    trunk_max    trunk_median    trunk_min     trunk_range      Urine.Ph_max  Urine.Ph_median
##  Min.   :0.0   Min.   :0.00   Min.   :0.00   Min.   :0.0000   Min.   :5.00   Min.   :5.00
##  1st Qu.:5.0   1st Qu.:3.00   1st Qu.:1.00   1st Qu.:0.0036   1st Qu.:6.00   1st Qu.:5.00
##  Median :7.0   Median :5.00   Median :3.00   Median :0.0069   Median :7.00   Median :6.00
##  Mean   :6.2   Mean   :4.89   Mean   :2.96   Mean   :0.0071   Mean   :6.82   Mean   :5.71
##  3rd Qu.:8.0   3rd Qu.:6.50   3rd Qu.:5.00   3rd Qu.:0.0096   3rd Qu.:7.00   3rd Qu.:6.00
##  Max.   :8.0   Max.   :8.00   Max.   :8.00   Max.   :0.0420   Max.   :9.00   Max.   :9.00
##   Urine.Ph_min
##  Min.   :5.00
##  1st Qu.:5.00
##  Median :5.00
##  Mean   :5.18
##  3rd Qu.:5.00
##  Max.   :8.00
```

- diverse variables
- multiple features are highly correlated
- some of variables represent statistics like `max`, `min` and `median` values of the same clinical measurements

Boruta()

https://mbq.github.io/Boruta/

- reference: https://cran.r-project.org/web/packages/Boruta/Boruta.pdf
- vignette: https://cran.r-project.org/web/packages/Boruta/vignettes/inahurry.pdf
- detailed methodology: https://www.jstatsoft.org/article/view/v036i11

Overview of boruta algorithm:

- adds randomness to data by creating shuffled copies of all features (shadow features);
- train a random forest on the extended data set to compute feature importance;
- iteratively remove features that are less important than the best shadow features;
- stops when all features are confirmed or rejected or a specified limit of random forest runs is reached.

Note: This will take a few minutes to complete.

```
library(Boruta)
set.seed(43612)
als <- Boruta(ALSFRS_slope~.-ID, data=ALS.train, doTrace=0)
als
```

```
## Boruta performed 99 iterations in 2.04 mins.
##  28 attributes confirmed important: ALSFRS_Total_max, ALSFRS_Total_median,
## ALSFRS_Total_min, ALSFRS_Total_range, Creatinine_max and 23 more;
##  62 attributes confirmed unimportant: Age_mean, Albumin_max, Albumin_median, Albumin_min,
## Albumin_range and 57 more;
##  9 tentative attributes left: Hematocrit_median, Hematocrit_range, Hemoglobin_max,
## Hemoglobin_median, Hemoglobin_min and 4 more;
```

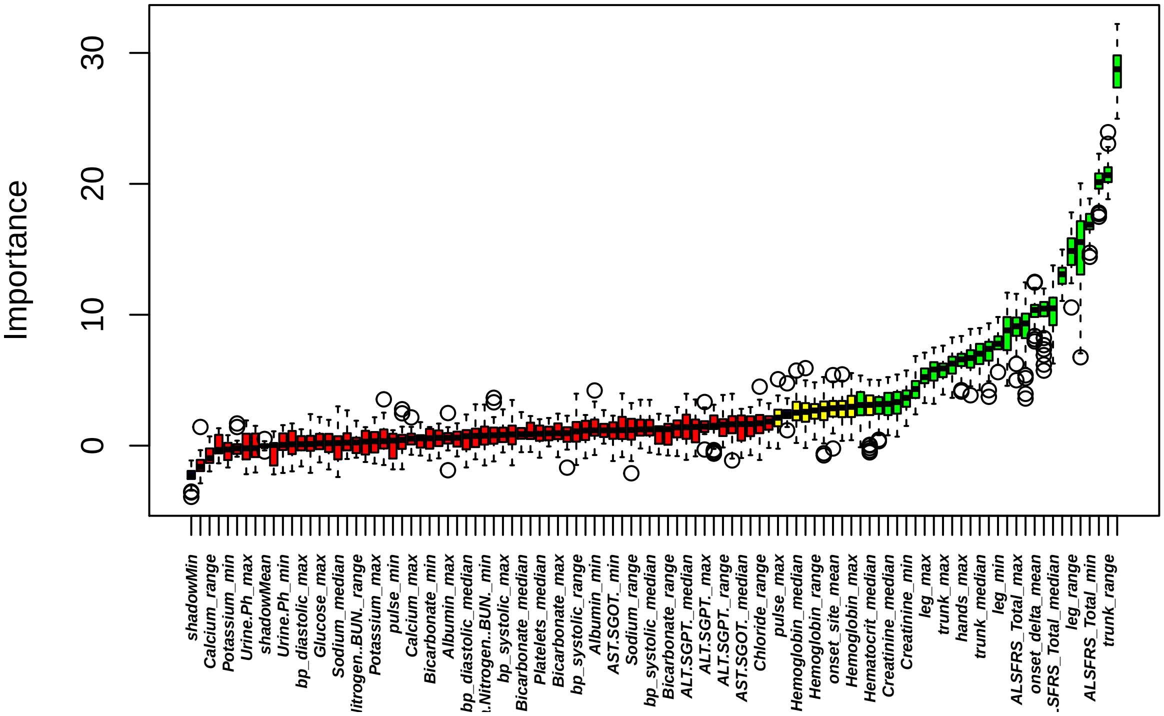
The importance scores for all features at every iteration are stored in the data frame `als$ImpHistory`.

```
als$ImpHistory[1:6, 1:10]
```

```
##      Age_mean Albumin_max Albumin_median Albumin_min Albumin_range ALSFRS_Total_max
## [1,]   0.4120       0.857          0.439       1.324          2.15             8.27
## [2,]   0.4091       0.436          1.250       1.167          2.90             7.66
## [3,]   0.1518       2.493          0.859       1.839          1.00             7.12
## [4,]   0.1460       0.257          2.468       2.547          1.91             8.62
## [5,]   1.8671       0.162          1.189       0.841          1.40             8.46
## [6,]  -0.0595       0.483          1.531      -0.239          1.63             8.34
##      ALSFRS_Total_median ALSFRS_Total_min ALSFRS_Total_range ALT.SGPT._max
## [1,]                6.27             15.0               25.2         1.500
## [2,]                8.26             16.6               26.0         2.924
## [3,]                6.82             15.4               26.3         1.804
## [4,]                8.24             16.7               25.5         1.039
## [5,]                8.70             16.3               26.3         1.304
## [6,]                8.00             16.8               25.2         0.887
```

A graph depicting the essential features.

```
plot(als, xlab="", xaxt="n")
lz<-lapply(1:ncol(als$ImpHistory), function(i)
als$ImpHistory[is.finite(als$ImpHistory[, i]), i])
names(lz)<-colnames(als$ImpHistory)
lb<-sort(sapply(lz, median))
axis(side=1, las=2, labels=names(lb), at=1:ncol(als$ImpHistory), cex.axis=0.5, font = 4)
```



Variables with green boxes are more important than the ones represented with red boxes, and we can see the range of importance scores within a single variable in the graph.

It may be desirable to get rid of tentative features.

- use only when a strict decision is highly desired.

```
final.als<-TentativeRoughFix(als)
final.als
```

```
## Boruta performed 99 iterations in 2.04 mins.
## Tentatives roughfixed over the last 99 iterations.
##  30 attributes confirmed important: ALSFRS_Total_max, ALSFRS_Total_median,
## ALSFRS_Total_min, ALSFRS_Total_range, Creatinine_max and 25 more;
##  69 attributes confirmed unimportant: Age_mean, Albumin_max, Albumin_median, Albumin_min,
## Albumin_range and 64 more;
```

```
final.als$finalDecision
```

```
##                         Age_mean                      Albumin_max                   Albumin_median 
##                         Rejected                         Rejected                         Rejected 
##                      Albumin_min                    Albumin_range                 ALSFRS_Total_max 
##                         Rejected                         Rejected                        Confirmed 
##              ALSFRS_Total_median                 ALSFRS_Total_min               ALSFRS_Total_range 
##                        Confirmed                        Confirmed                        Confirmed 
##                    ALT.SGPT._max                 ALT.SGPT._median                    ALT.SGPT._min 
##                         Rejected                         Rejected                         Rejected 
##                  ALT.SGPT._range                    AST.SGOT._max                 AST.SGOT._median 
##                         Rejected                         Rejected                         Rejected 
##                    AST.SGOT._min                  AST.SGOT._range                  Bicarbonate_max 
##                         Rejected                         Rejected                         Rejected 
##               Bicarbonate_median                  Bicarbonate_min                Bicarbonate_range 
##                         Rejected                         Rejected                         Rejected 
##    Blood.Urea.Nitrogen..BUN._max Blood.Urea.Nitrogen..BUN._median    Blood.Urea.Nitrogen..BUN._min 
##                         Rejected                         Rejected                         Rejected 
##  Blood.Urea.Nitrogen..BUN._range                 bp_diastolic_max              bp_diastolic_median 
##                         Rejected                         Rejected                         Rejected 
##                 bp_diastolic_min               bp_diastolic_range                  bp_systolic_max 
##                         Rejected                         Rejected                         Rejected 
##               bp_systolic_median                  bp_systolic_min                bp_systolic_range 
##                         Rejected                         Rejected                         Rejected 
##                      Calcium_max                   Calcium_median                      Calcium_min 
##                         Rejected                         Rejected                         Rejected 
##                    Calcium_range                     Chloride_max                  Chloride_median 
##                         Rejected                         Rejected                         Rejected 
##                     Chloride_min                   Chloride_range                   Creatinine_max 
##                         Rejected                         Rejected                        Confirmed 
##                Creatinine_median                   Creatinine_min                 Creatinine_range 
##                        Confirmed                        Confirmed                         Rejected 
##                      Gender_mean                      Glucose_max                   Glucose_median 
##                         Rejected                         Rejected                         Rejected 
##                      Glucose_min                    Glucose_range                        hands_max 
##                         Rejected                         Rejected                        Confirmed 
##                     hands_median                        hands_min                      hands_range 
##                        Confirmed                        Confirmed                        Confirmed 
##                   Hematocrit_max                Hematocrit_median                   Hematocrit_min 
##                        Confirmed                         Rejected                        Confirmed 
##                 Hematocrit_range                   Hemoglobin_max                Hemoglobin_median 
##                         Rejected                         Rejected                         Rejected 
##                   Hemoglobin_min                 Hemoglobin_range                          leg_max 
##                        Confirmed                         Rejected                        Confirmed 
##                       leg_median                          leg_min                        leg_range 
##                        Confirmed                        Confirmed                        Confirmed 
##                        mouth_max                     mouth_median                        mouth_min 
##                        Confirmed                        Confirmed                        Confirmed 
##                      mouth_range                 onset_delta_mean                  onset_site_mean 
##                        Confirmed                        Confirmed                         Rejected 
##                    Platelets_max                 Platelets_median                    Platelets_min 
##                         Rejected                         Rejected                         Rejected 
##                    Potassium_max                 Potassium_median                    Potassium_min 
##                         Rejected                         Rejected                         Rejected
##                   Potassium_range                        pulse_max                     pulse_median
##                          Rejected                        Confirmed                         Rejected
##                         pulse_min                      pulse_range                  respiratory_max
##                          Rejected                         Rejected                         Rejected
##                respiratory_median                  respiratory_min                respiratory_range
##                          Rejected                        Confirmed                        Confirmed
##                        Sodium_max                    Sodium_median                       Sodium_min
##                          Rejected                         Rejected                         Rejected
##                      Sodium_range                        SubjectID                        trunk_max
##                          Rejected                         Rejected                        Confirmed
##                      trunk_median                        trunk_min                      trunk_range
##                         Confirmed                        Confirmed                        Confirmed
##                      Urine.Ph_max                  Urine.Ph_median                     Urine.Ph_min
##                          Rejected                         Rejected                         Rejected
## Levels: Tentative Confirmed Rejected
```

Report the Boruta “Confirmed” & “Tentative” features, removing the “Rejected” ones

```
print(final.als$finalDecision[final.als$finalDecision %in% c("Confirmed", "Tentative")])
```

```
##     ALSFRS_Total_max  ALSFRS_Total_median     ALSFRS_Total_min   ALSFRS_Total_range       Creatinine_max
##            Confirmed            Confirmed            Confirmed            Confirmed            Confirmed
##    Creatinine_median       Creatinine_min            hands_max         hands_median            hands_min
##            Confirmed            Confirmed            Confirmed            Confirmed            Confirmed
##          hands_range       Hematocrit_max       Hematocrit_min       Hemoglobin_min              leg_max
##            Confirmed            Confirmed            Confirmed            Confirmed            Confirmed
##           leg_median              leg_min            leg_range            mouth_max         mouth_median
##            Confirmed            Confirmed            Confirmed            Confirmed            Confirmed
##            mouth_min          mouth_range     onset_delta_mean            pulse_max      respiratory_min
##            Confirmed            Confirmed            Confirmed            Confirmed            Confirmed
##    respiratory_range            trunk_max         trunk_median            trunk_min          trunk_range
##            Confirmed            Confirmed            Confirmed            Confirmed            Confirmed
## Levels: Tentative Confirmed Rejected
```

Recursive feature elimination (RFE)

Recursive feature elimination (RFE) is basically a backward selection.

- build a model on the entire set of variables
- compute an importance score for each variables
- remove the least important variable(s) from the model
- re-build a model and re-compute importance scores
- ...

The subset size is a tuning parameter for RFE

- the subset size that optimizes the loss is used to select the variables based on the importance rankings;
- the optimal subset is then used to train the final model.

RFE is frequently used with random forest models

- random forest tends not to exclude variables
- internal method for measuring variable importance

Measuring variable importance in suffers from multicollinearity. When there are highly correlated variables in a training set that are useful for predicting the outcome, then which variable is chosen for partitioning the samples is essentially a random selection.

- dilutes the importance scores
- it might be beneficial to filter out highly correlated features

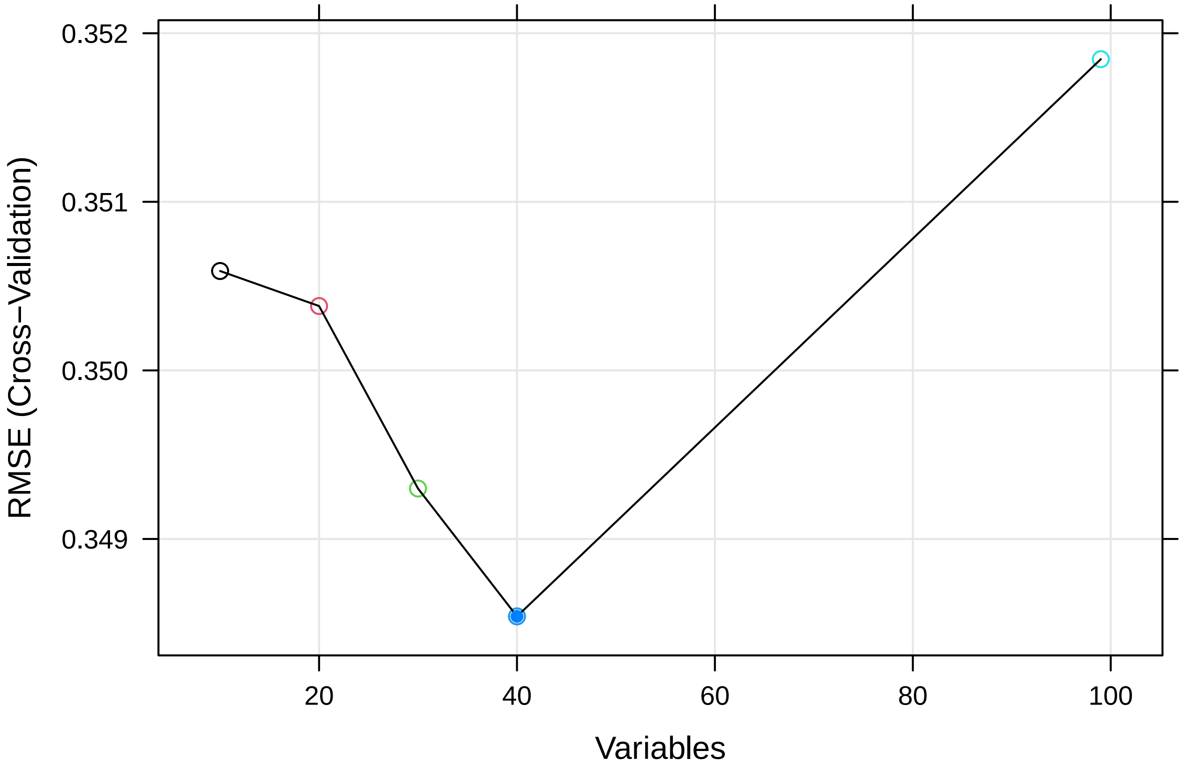
Note: This will take a few minutes to complete.

```
library(caret)
library(randomForest)
set.seed(43612)
control <- rfeControl(functions = rfFuncs, method = "cv", number=10)
rf.train <- rfe(ALS.train[, -c(1, 7)], ALS.train[, 7],
                sizes=c(10, 20, 30, 40), rfeControl=control)
rf.train
```

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (10 fold)
##
## Resampling performance over subset size:
##
##  Variables  RMSE Rsquared   MAE RMSESD RsquaredSD  MAESD Selected
##         10 0.351    0.684 0.249 0.0364     0.0586 0.0201
##         20 0.350    0.685 0.248 0.0366     0.0559 0.0197
##         30 0.349    0.687 0.248 0.0343     0.0546 0.0194
##         40 0.349    0.689 0.248 0.0339     0.0524 0.0186        *
##         99 0.352    0.682 0.250 0.0343     0.0539 0.0176
##
## The top 5 variables (out of 40):
##    ALSFRS_Total_range, trunk_range, hands_range, mouth_range, ALSFRS_Total_min
```

- the `sizes=` option allows us to specify the number of variables we want to include in the model

```
plot(rf.train, type=c("g", "o"), cex=1, col=1:5)
```



Common variables chosen by the two techniques:

```
predRFE <- predictors(rf.train)
predBoruta <- getSelectedAttributes(final.als, withTentative = F)
intersect(predBoruta, predRFE)
```

```
##  [1] "ALSFRS_Total_max"    "ALSFRS_Total_median" "ALSFRS_Total_min"    "ALSFRS_Total_range"
##  [5] "Creatinine_max"      "Creatinine_median"   "Creatinine_min"      "hands_max"
##  [9] "hands_median"        "hands_min"           "hands_range"         "Hematocrit_max"
## [13] "Hemoglobin_min"      "leg_max"             "leg_median"          "leg_min"
## [17] "leg_range"           "mouth_median"        "mouth_min"           "mouth_range"
## [21] "onset_delta_mean"    "respiratory_min"     "respiratory_range"   "trunk_max"
## [25] "trunk_median"        "trunk_min"           "trunk_range"
```

Example: Simulation Study

Let us also investigate the two procedures on a simulation benchmark where we know the true variable. We are going use the "Friedman 1" benchmark (Friedman, 1991; Breiman, 1996). Inputs are 10 independent variables uniformly distributed on the interval $[0, 1]$, only 5 out of these 10 are actually related to the outputs. Outputs are created according to the formula

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

We use the `mlbench` library to obtain the data.

```
library(mlbench)
n <- 100
p <- 40
sigma <- 1
set.seed(4125)
sim <- mlbench.friedman1(n, sd = sigma)
colnames(sim$x) <- c(paste("real", 1:5, sep = ""),
                     paste("bogus", 1:5, sep = ""))
bogus <- matrix(rnorm(n * p), nrow = n)
colnames(bogus) <- paste("bogus", 5+(1:ncol(bogus)), sep = "")
x <- cbind(sim$x, bogus)
y <- sim$y
```

We added 40 additional pure noise variables that are univariate standard normals.

The predictors are centered and scaled:

```
normalization <- preProcess(x)
x <- predict(normalization, x)
x <- as.data.frame(x)
subsets <- c(1:5, 10, 15, 20, 25)
```

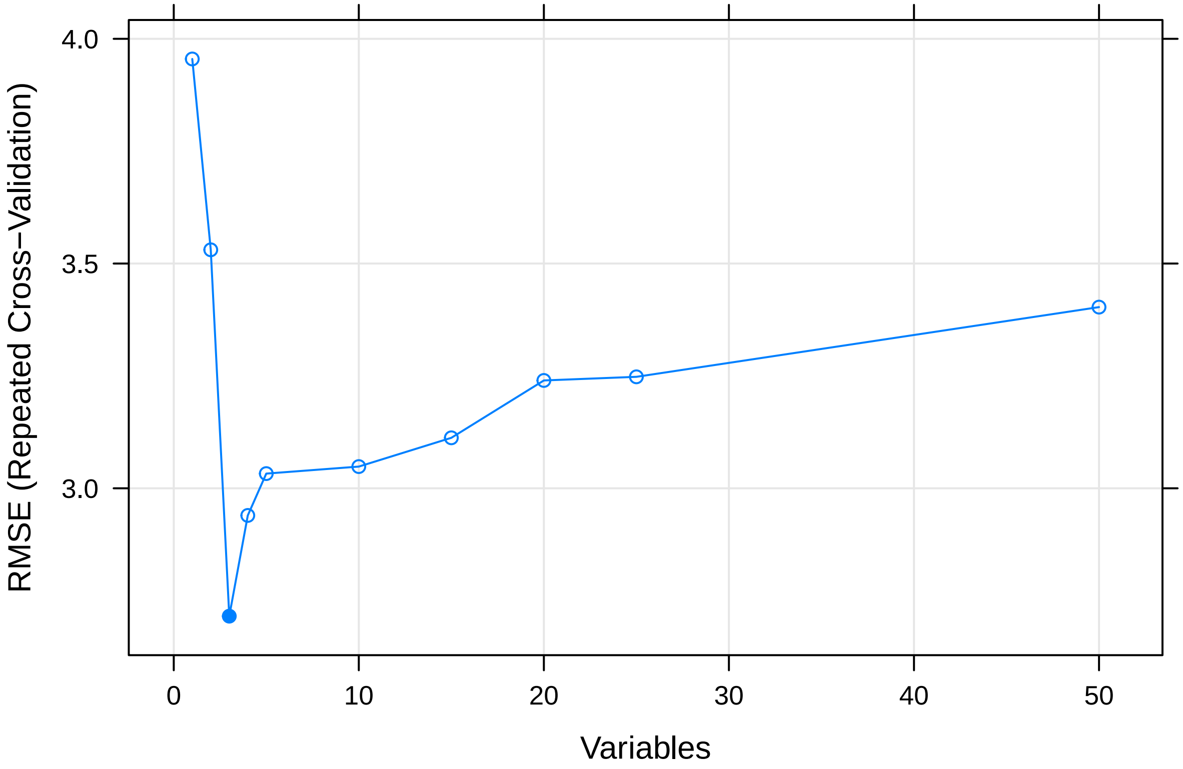
The simulation will fit models with subset sizes of 25, 20, 15, 10, 5, 4, 3, 2, 1.

Note: This will take a few minutes to complete.

```
control <- rfeControl(functions = rfFuncs, method = "repeatedcv", number=10, repeats = 5)
rf.sim  <- rfe(x, y, sizes=subsets, rfeControl=control)
rf.sim
```

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (10 fold, repeated 5 times)
##
## Resampling performance over subset size:
##
##  Variables RMSE Rsquared  MAE RMSESD RsquaredSD MAESD Selected
##          1 3.96    0.409 3.15  0.690      0.224 0.653
##          2 3.53    0.519 2.89  0.535      0.177 0.505
##          3 2.72    0.743 2.22  0.434      0.116 0.368        *
##          4 2.94    0.721 2.41  0.484      0.156 0.448
##          5 3.03    0.724 2.50  0.485      0.140 0.432
##         10 3.05    0.738 2.52  0.470      0.142 0.409
##         15 3.11    0.722 2.55  0.471      0.146 0.389
##          20 3.24    0.715 2.65  0.496      0.142 0.392
##          25 3.25    0.714 2.66  0.470      0.137 0.376
##          50 3.40    0.707 2.79  0.478      0.134 0.384
##
## The top 3 variables (out of 3):
##    real4, real2, real1
```

```
plot(rf.sim, type = c("g", "o"))
```



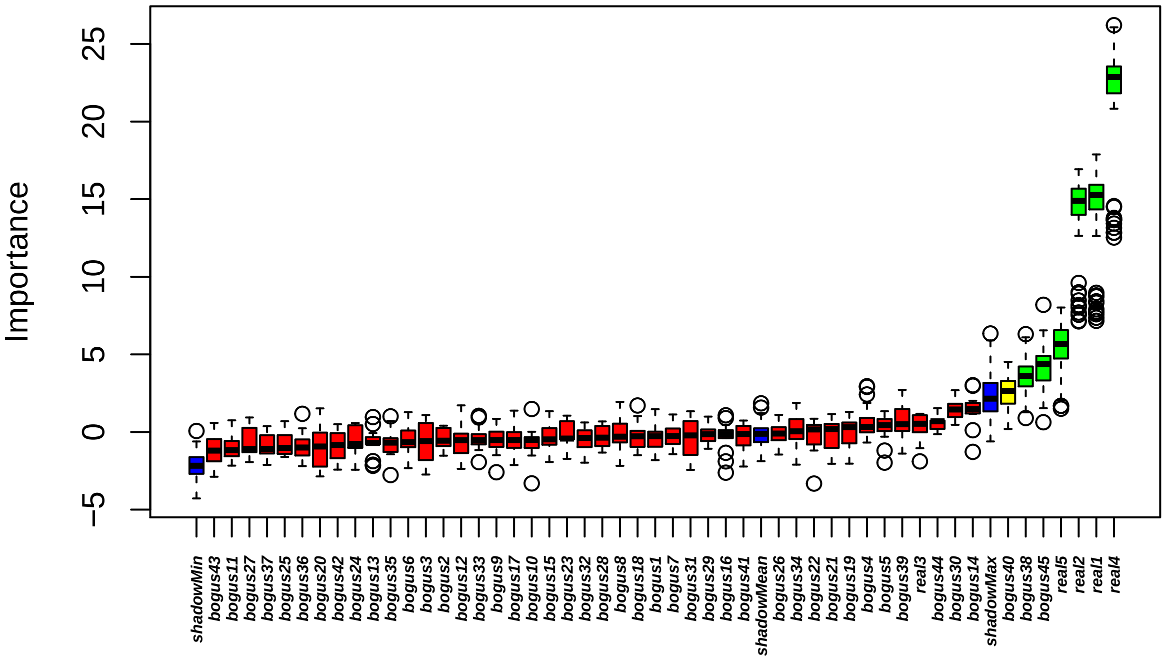
Note: This will take a few minutes to complete.

```
boruta.sim <- Boruta(x, y)
boruta.sim
```

```
## Boruta performed 99 iterations in 1.52 secs.
##  6 attributes confirmed important: bogus38, bogus45, real1, real2, real4 and 1 more;
##  43 attributes confirmed unimportant: bogus1, bogus10, bogus11, bogus12, bogus13 and 38
## more;
##  1 tentative attributes left: bogus40;
```

Result plot

```
plot(boruta.sim, xlab="", xaxt="n")
lz<-lapply(1:ncol(boruta.sim$ImpHistory), function(i)
boruta.sim$ImpHistory[is.finite(boruta.sim$ImpHistory[, i]), i])
names(lz)<-colnames(boruta.sim$ImpHistory)
lb<-sort(sapply(lz, median))
axis(side=1, las=2, labels=names(lb), at=1:ncol(boruta.sim$ImpHistory), cex.axis=0.5, font = 4)
```



Attribute statistics

```
attStats(boruta.sim)
```

```
##          meanImp medianImp minImp maxImp normHits  decision
## real1    14.4168   15.2651  7.165 17.884   1.0000 Confirmed
## real2    14.1189   14.8900  7.127 16.929   1.0000 Confirmed
## real3     0.2871    0.5332 -1.892  1.176   0.0000  Rejected
## real4    21.7585   22.8692 12.530 26.208   1.0000 Confirmed
## real5     5.4911    5.6776  1.501  8.020   0.8990 Confirmed
## bogus1   -0.4122   -0.2757 -1.812  1.468   0.0000  Rejected
## bogus2   -0.4758   -0.5583 -1.525  0.408   0.0000  Rejected
## bogus3   -0.6822   -0.5854 -2.740  1.093   0.0000  Rejected
## bogus4    0.6139    0.3295 -0.679  2.950   0.0202  Rejected
## bogus5    0.2790    0.4434 -1.974  1.337   0.0000  Rejected
## bogus6   -0.5103   -0.6569 -2.328  1.277   0.0000  Rejected
## bogus7   -0.1858   -0.2598 -1.428  1.128   0.0000  Rejected
## bogus8   -0.1147   -0.2987 -2.176  1.938   0.0000  Rejected
## bogus9   -0.5360   -0.5172 -2.591  0.850   0.0000  Rejected
## bogus10  -0.6803   -0.5030 -3.309  1.482   0.0000  Rejected
## bogus11  -1.0344   -1.1706 -2.166  0.751   0.0000  Rejected
## bogus12  -0.5189   -0.5325 -2.371  1.715   0.0000  Rejected
## bogus13  -0.6902   -0.7050 -2.178  0.962   0.0000  Rejected
## bogus14   1.4234    1.4774 -1.279  3.015   0.0000  Rejected
## bogus15  -0.3233   -0.4783 -1.932  1.338   0.0000  Rejected
## bogus16  -0.3419   -0.1283 -2.620  1.076   0.0000  Rejected
## bogus17  -0.5127   -0.5152 -2.126  1.377   0.0000  Rejected
## bogus18  -0.2664   -0.2824 -1.496  1.708   0.0000  Rejected
## bogus19  -0.0180    0.3102 -2.039  1.296   0.0000  Rejected
## bogus20  -0.8763   -0.9344 -2.861  1.520   0.0000  Rejected
## bogus21  -0.1871    0.1884 -2.046  1.154   0.0000  Rejected
## bogus22  -0.2004    0.1631 -3.322  0.856   0.0000  Rejected
## bogus23  -0.0901   -0.4230 -1.726  1.055   0.0000  Rejected
## bogus24  -0.6113   -0.7626 -2.431  0.578   0.0000  Rejected
## bogus25 -0.7665   -1.0224 -1.605  0.693   0.0000  Rejected
## bogus26 -0.1394   -0.0987 -1.453  1.102   0.0000  Rejected
## bogus27 -0.6906   -1.0923 -1.938  0.936   0.0000  Rejected
## bogus28 -0.3204   -0.3632 -1.325  0.680   0.0000  Rejected
## bogus29 -0.0660   -0.1477 -1.078  0.993   0.0000  Rejected
## bogus30  1.4097    1.4494  0.467  2.689   0.0000  Rejected
## bogus31 -0.4300   -0.2220 -2.440  1.334   0.0000  Rejected
## bogus32 -0.4727   -0.3760 -1.975  0.614   0.0000  Rejected
## bogus33 -0.3896   -0.5270 -1.939  1.042   0.0000  Rejected
## bogus34  0.0572    0.0412 -2.102  1.876   0.0000  Rejected
## bogus35 -0.7085   -0.6838 -2.773  1.010   0.0000  Rejected
## bogus36 -0.8124   -1.0011 -2.196  1.172   0.0000  Rejected
## bogus37 -0.8787   -1.0829 -2.120  0.375   0.0000  Rejected
## bogus38  3.5504    3.6036  0.900  6.305   0.7778 Confirmed
## bogus39  0.6811    0.4962 -1.391  2.714   0.0303  Rejected
## bogus40  2.5564    2.6559  0.195  4.521   0.6162 Tentative
## bogus41 -0.3359   -0.1254 -2.225  0.738   0.0000  Rejected
## bogus42 -0.9072   -0.8360 -2.424  0.498   0.0000  Rejected
## bogus43 -1.0997   -1.1976 -2.881  0.585   0.0000  Rejected
## bogus44  0.5732    0.6881 -0.140  1.535   0.0000  Rejected
## bogus45  4.1863    4.3719  0.630  8.197   0.8182 Confirmed
```

Let us now try with a slightly large sample size $n = 200$. Everything else is the same.

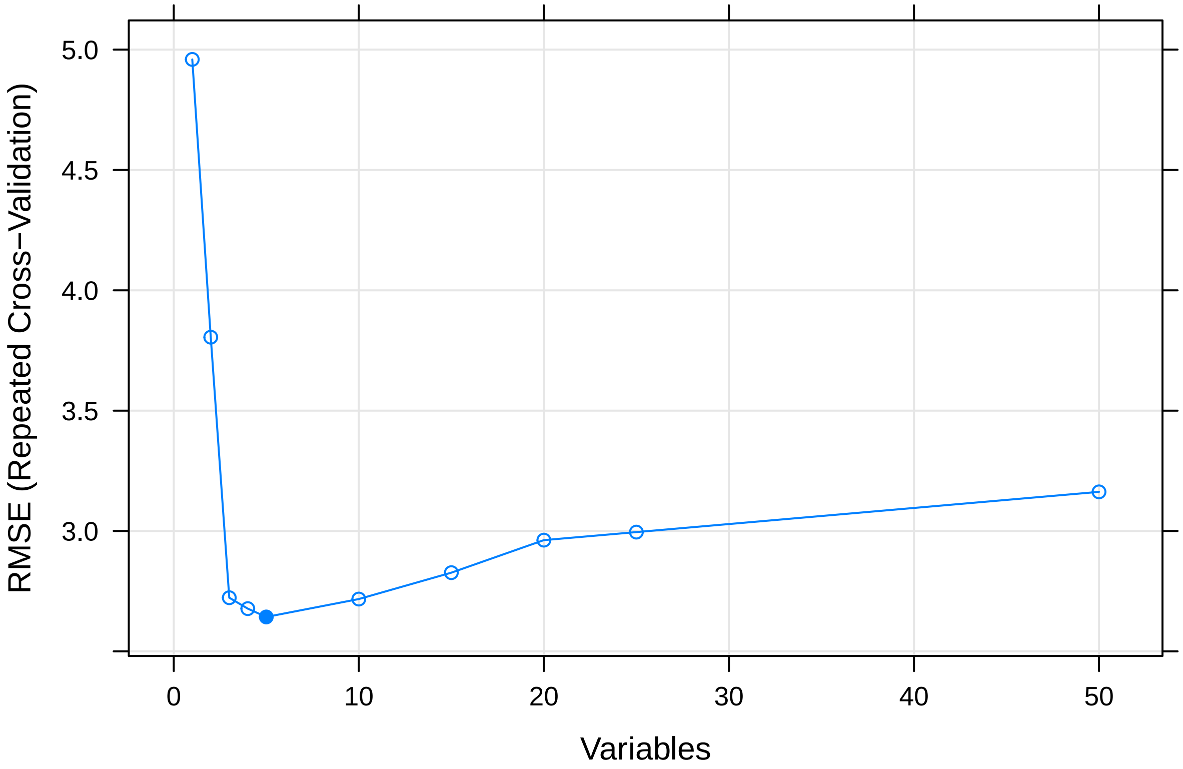
Note: This will take a few minutes to complete.

```
n <- 200
p <- 40
sigma <- 1
set.seed(4125)
sim <- mlbench.friedman1(n, sd = sigma)
colnames(sim$x) <- c(paste("real", 1:5, sep = ""),
                     paste("bogus", 1:5, sep = ""))
bogus <- matrix(rnorm(n * p), nrow = n)
colnames(bogus) <- paste("bogus", 5+(1:ncol(bogus)), sep = "")
x <- cbind(sim$x, bogus)
y <- sim$y
normalization <- preProcess(x)
x <- predict(normalization, x)
x <- as.data.frame(x)
subsets <- c(1:5, 10, 15, 20, 25)

control <- rfeControl(functions = rfFuncs, method = "repeatedcv", number=10, repeats = 5)
rf.sim  <- rfe(x, y, sizes=subsets, rfeControl=control)
rf.sim
```

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (10 fold, repeated 5 times)
##
## Resampling performance over subset size:
##
##   Variables RMSE Rsquared  MAE RMSESD RsquaredSD MAESD Selected
##           1 4.96   0.0732 4.05  0.707     0.0760 0.587
##           2 3.81   0.3352 3.12  0.479     0.1530 0.451
##          3 2.72   0.6762 2.17  0.376     0.1120 0.325
##          4 2.68   0.7201 2.20  0.371     0.1176 0.348
##          5 2.64   0.7890 2.15  0.391     0.0834 0.329        *
##         10 2.72   0.7698 2.20  0.380     0.0834 0.311
##         15 2.83   0.7399 2.29  0.414     0.0978 0.343
##         20 2.96   0.7269 2.41  0.422     0.1028 0.339
##         25 3.00   0.7166 2.43  0.433     0.1102 0.353
##         50 3.16   0.6810 2.57  0.423     0.1214 0.335
##
## The top 5 variables (out of 5):
##    real1, real2, real4, real5, real3
```

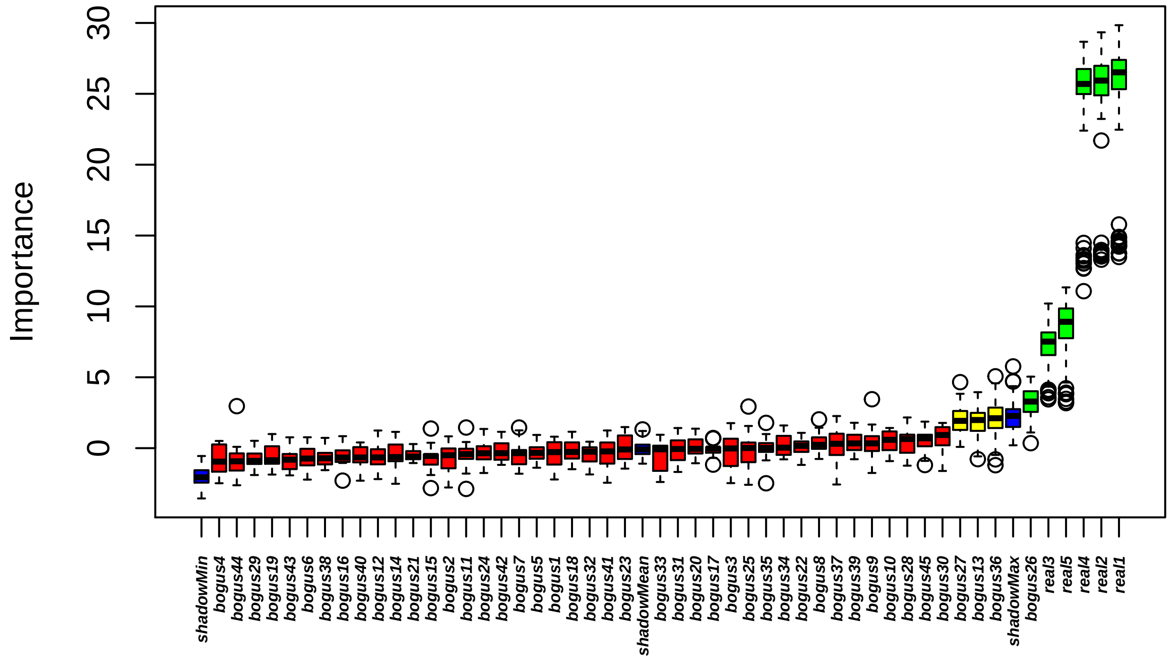
```
plot(rf.sim, type = c("g", "o"))
```



```
boruta.sim <- Boruta(x, y)
boruta.sim
```

```
## Boruta performed 99 iterations in 3.36 secs.
##  6 attributes confirmed important: bogus26, real1, real2, real3, real4 and 1 more;
##  41 attributes confirmed unimportant: bogus1, bogus10, bogus11, bogus12, bogus14 and 36
## more;
##  3 tentative attributes left: bogus13, bogus27, bogus36;
```

```
plot(boruta.sim, xlab="", xaxt="n")
lz<-lapply(1:ncol(boruta.sim$ImpHistory), function(i)
boruta.sim$ImpHistory[is.finite(boruta.sim$ImpHistory[, i]), i])
names(lz)<-colnames(boruta.sim$ImpHistory)
lb<-sort(sapply(lz, median))
axis(side=1, las=2, labels=names(lb), at=1:ncol(boruta.sim$ImpHistory), cex.axis=0.5, font = 4)
```



```
attStats(boruta.sim)
```

```
##             meanImp medianImp   minImp maxImp normHits  decision
## real1   25.053076   26.5155 13.4857 29.840   1.0000 Confirmed
## real2   24.560321   25.9342 13.2992 29.340   1.0000 Confirmed
## real3    7.220585    7.5180  3.4442 10.206   0.9899 Confirmed
## real4   24.368496   25.6965 11.0746 28.673   1.0000 Confirmed
## real5    8.566450    8.9142  3.1754 11.342   0.9899 Confirmed
## bogus1  -0.404292   -0.2789 -2.2098  0.809   0.0000  Rejected
## bogus2  -0.832059   -0.4985 -2.7748  0.833   0.0000  Rejected
## bogus3  -0.267769   -0.0209 -2.4676  1.766   0.0000  Rejected
## bogus4  -0.827315   -0.9488 -2.4737  0.507   0.0000  Rejected
## bogus5  -0.282409   -0.3281 -1.3830  0.931   0.0000  Rejected
## bogus6  -0.653863   -0.7346 -2.2195  0.768   0.0000  Rejected
## bogus7  -0.386086   -0.3479 -1.8040  1.454   0.0000  Rejected
## bogus8   0.415413    0.2368 -0.7636  2.029   0.0000  Rejected
## bogus9   0.375869    0.3359 -1.7517  3.450   0.0101  Rejected
## bogus10  0.471256    0.5885 -0.9141  1.424   0.0000  Rejected
## bogus11 -0.466887   -0.4217 -2.8748  1.454   0.0000  Rejected
## bogus12 -0.585345   -0.6544 -2.1831  1.244   0.0000  Rejected
## bogus13  1.776393    1.9851 -0.7756  3.950   0.4242 Tentative
## bogus14 -0.482675   -0.6288 -2.5130  1.148   0.0000  Rejected
## bogus15 -0.745976   -0.5519 -2.8194  1.381   0.0000  Rejected
## bogus16 -0.548214   -0.6603 -2.2924  0.845   0.0000  Rejected
## bogus17 -0.075336   -0.0288 -1.1625  0.711   0.0000  Rejected
## bogus18 -0.241221   -0.2671 -1.4838  1.158   0.0000  Rejected
## bogus19 -0.626955   -0.8546 -1.8638  0.995   0.0000  Rejected
## bogus20  0.077822   -0.0491 -1.0636  1.374   0.0000  Rejected
## bogus21 -0.502531   -0.5947 -1.0513  0.289   0.0000  Rejected
## bogus22  0.124647    0.1656 -1.1825  1.080   0.0000  Rejected
## bogus23 -0.000906   -0.0929 -1.4442  1.487   0.0000  Rejected
## bogus24 -0.335973   -0.3608 -1.7523  1.353   0.0000  Rejected
## bogus25 -0.034877    0.0117 -2.5847  2.935   0.0101  Rejected
## bogus26  3.238152    3.2876  0.3516  5.045   0.7677 Confirmed
## bogus27  2.006880    1.9267  0.0885  4.655   0.4040 Tentative
## bogus28  0.393491    0.6451 -1.2355  2.166   0.0000  Rejected
## bogus29 -0.780894   -0.8982 -1.8962  0.520   0.0000  Rejected
## bogus30  0.728078    0.9278 -1.6100  1.778   0.0000  Rejected
## bogus31 -0.166516   -0.0602 -1.6803  1.419   0.0000  Rejected
## bogus32 -0.431643   -0.2485 -1.8541  0.456   0.0000  Rejected
## bogus33 -0.470244   -0.0833 -2.3879  0.945   0.0000  Rejected
## bogus34  0.171497    0.0330 -0.7908  1.600   0.0000  Rejected
## bogus35  0.008024    0.0151 -2.4785  1.783   0.0000  Rejected
## bogus36  2.066192    2.1096 -1.2037  5.068   0.4646 Tentative
## bogus37  0.296761    0.3103 -2.5647  2.266   0.0000  Rejected
## bogus38 -0.589869   -0.7163 -1.5509  0.725   0.0000  Rejected
## bogus39  0.450585    0.3356 -0.7814  1.791   0.0000  Rejected
## bogus40 -0.627772   -0.6582 -2.2954  0.403   0.0000  Rejected
## bogus41 -0.279155   -0.2293 -2.4399  1.248   0.0000  Rejected
## bogus42 -0.228526   -0.3534 -1.1796  1.154   0.0000  Rejected
## bogus43 -0.769450   -0.8143 -1.9202  0.765   0.0000  Rejected
## bogus44 -0.750402   -0.9276 -2.6177  2.963   0.0000  Rejected
## bogus45  0.371304    0.6918 -1.1921  1.871   0.0000  Rejected
```