Decision tree and random forest

Chuxuan He

2022-11-19

```
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.1.3
```

```
## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(DAAG)
```

```
## Warning: package 'DAAG' was built under R version 4.1.3
```

```
library(party)
```

```
## Warning: package 'party' was built under R version 4.1.3
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Warning: package 'mvtnorm' was built under R version 4.1.1
```

```
##
## Attaching package: 'mvtnorm'
```

```
## The following object is masked from 'package:mclust':
##
##     dmvnorm
```

```
## Loading required package: modeltools
```

```
## Warning: package 'modeltools' was built under R version 4.1.1
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Warning: package 'strucchange' was built under R version 4.1.3
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.1.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Warning: package 'sandwich' was built under R version 4.1.3
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.1.3
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.1.3
```

```
library(mlbench)
```

```
## Warning: package 'mlbench' was built under R version 4.1.3
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.1.3
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.1.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

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

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

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

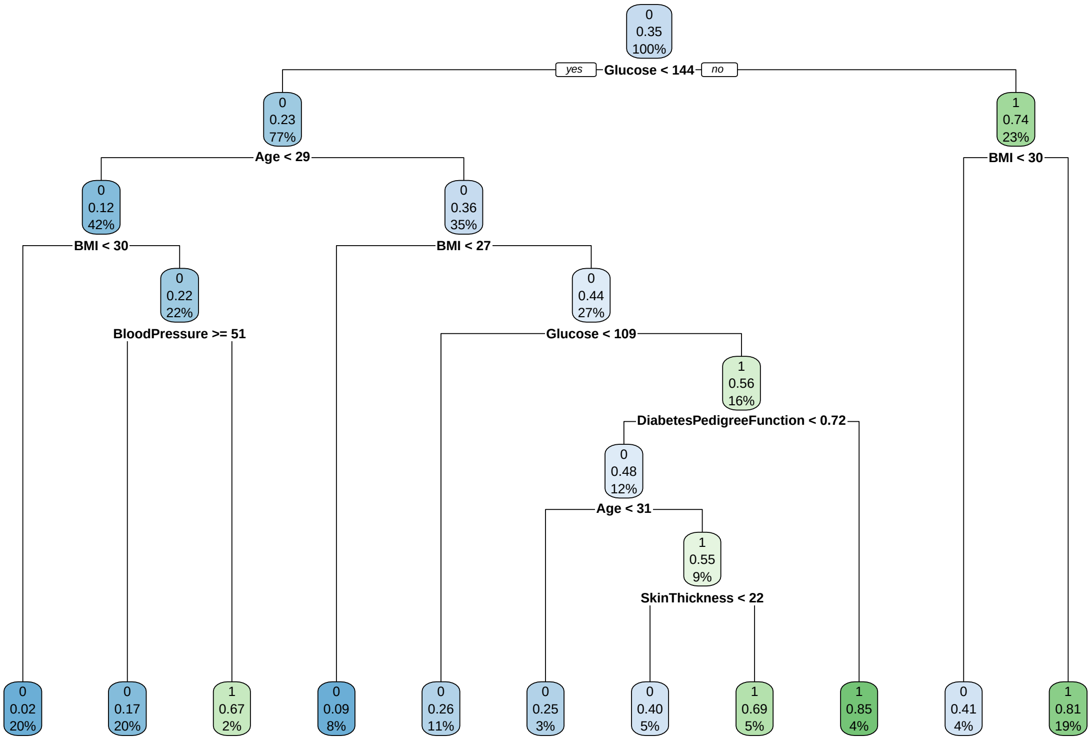
Data Cleaning

```
Diabetes<-read.csv("diabetes.csv")
set.seed(123)
trainIndex <- createDataPartition(Diabetes$Outcome, p = 0.7, list = FALSE)
bn.training <- Diabetes[trainIndex,]
bn.test <- Diabetes[-trainIndex,]
head(bn.training)
```

```
##    Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 2            1      85            66            29       0 26.6
## 5            0     137            40            35     168 43.1
## 6            5     116            74             0       0 25.6
## 7            3      78            50            32      88 31.0
## 8           10     115             0             0       0 35.3
## 10           8     125            96             0       0  0.0
##    DiabetesPedigreeFunction Age Outcome
## 2                     0.351  31       0
## 5                     2.288  33       1
## 6                     0.201  30       0
## 7                     0.248  26       1
## 8                     0.134  29       0
## 10                    0.232  54       1
```

tree

```
tree <- rpart(Outcome ~., data = bn.training,method='class',minsplit=10,minbucket=10)
rpart.plot(tree)
```



```
tree
```

```
## n= 538
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 538 187 0 (0.65241636 0.34758364)
##     2) Glucose< 143.5 415  96 0 (0.76867470 0.23132530)
##       4) Age< 28.5 227  28 0 (0.87665198 0.12334802)
##         8) BMI< 29.95 107   2 0 (0.98130841 0.01869159) *
##         9) BMI>=29.95 120  26 0 (0.78333333 0.21666667)
##          18) BloodPressure>=51 108  18 0 (0.83333333 0.16666667) *
##          19) BloodPressure< 51 12   4 1 (0.33333333 0.66666667) *
##       5) Age>=28.5 188  68 0 (0.63829787 0.36170213)
##        10) BMI< 27.05 43   4 0 (0.90697674 0.09302326) *
##        11) BMI>=27.05 145  64 0 (0.55862069 0.44137931)
##          22) Glucose< 108.5 58  15 0 (0.74137931 0.25862069) *
##          23) Glucose>=108.5 87  38 1 (0.43678161 0.56321839)
##            46) DiabetesPedigreeFunction< 0.7205 67  32 0 (0.52238806 0.47761194)
##              92) Age< 30.5 16   4 0 (0.75000000 0.25000000) *
##              93) Age>=30.5 51  23 1 (0.45098039 0.54901961)
##               186) SkinThickness< 21.5 25  10 0 (0.60000000 0.40000000) *
##               187) SkinThickness>=21.5 26   8 1 (0.30769231 0.69230769) *
##            47) DiabetesPedigreeFunction>=0.7205 20   3 1 (0.15000000 0.85000000) *
##     3) Glucose>=143.5 123  32 1 (0.26016260 0.73983740)
##       6) BMI< 29.85 22   9 0 (0.59090909 0.40909091) *
##       7) BMI>=29.85 101  19 1 (0.18811881 0.81188119) *
```

Random Forests

```
set.seed(123)
oob_train_control <- trainControl(method="oob", classProbs = TRUE, savePredictions = TRUE)
forestfit <- train(Outcome ~., data = bn.training, method = 'rf', importance = FALSE,trControl = oob_tra
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
## Warning in train.default(x, y, weights = w, ...): cannnot compute class
## probabilities for regression
```

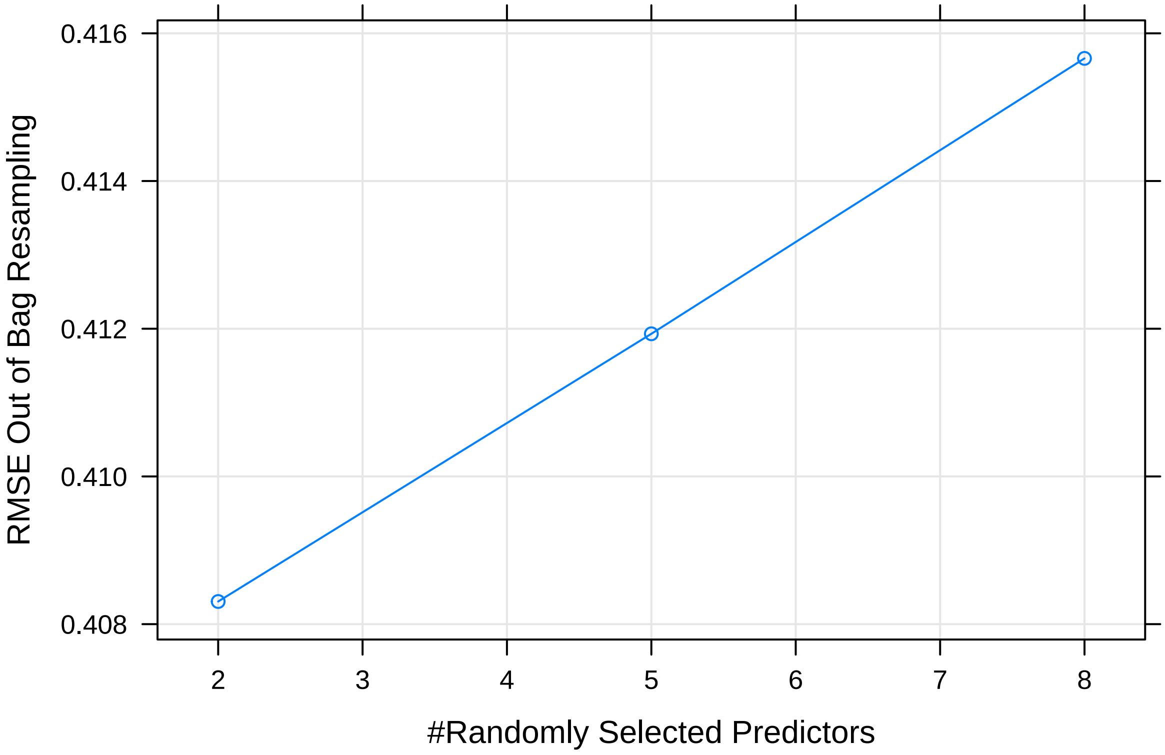
```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): The response has
## five or fewer unique values. Are you sure you want to do regression?

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): The response has
## five or fewer unique values. Are you sure you want to do regression?

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): The response has
## five or fewer unique values. Are you sure you want to do regression?

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): The response has
## five or fewer unique values. Are you sure you want to do regression?
```

```
plot(forestfit)
```



```
recommended.mtry <-floor(sqrt(ncol(bn.training)))
tunegrid <- expand.grid(mtry=recommended.mtry)
set.seed(123)
forestfit.m <- train(Outcome ~.,data = bn.training, method = 'rf', importance = FALSE, trControl = oob_
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.

## Warning in train.default(x, y, weights = w, ...): cannnot compute class
## probabilities for regression

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): The response has
## five or fewer unique values. Are you sure you want to do regression?

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): The response has
## five or fewer unique values. Are you sure you want to do regression?
```

```
print(forestfit.m, digits = 2)
```

```
## Random Forest
##
## 538 samples
##   8 predictor
```

```
##
## No pre-processing
## Resampling results:
##
##   RMSE  Rsquared
##   0.41  0.26
##
## Tuning parameter 'mtry' was held constant at a value of 3
```

```
set.seed(123)
forestfit.ntree <- train(Outcome ~.,data = bn.training, method = 'rf', ntree = 500, importance = T,trCo
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
## Warning in train.default(x, y, weights = w, ...): cannnot compute class
## probabilities for regression
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): The response has
## five or fewer unique values. Are you sure you want to do regression?
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): The response has
## five or fewer unique values. Are you sure you want to do regression?
```

```
print(forestfit.ntree, digits = 2)
```

```
## Random Forest
##
## 538 samples
##   8 predictor
##
## No pre-processing
## Resampling results:
##
##   RMSE  Rsquared
##   0.41  0.26
##
## Tuning parameter 'mtry' was held constant at a value of 3
```

```
forestfit.ntree$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, ntree = 500, mtry = param$mtry, importance = ..2)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 0.167706
##                     % Var explained: 26.05
```

```
library(randomForest)
```

```
## Warning: package ’randomForest’ was built under R version 4.1.3
```

```
## randomForest 4.7-1.1
```

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

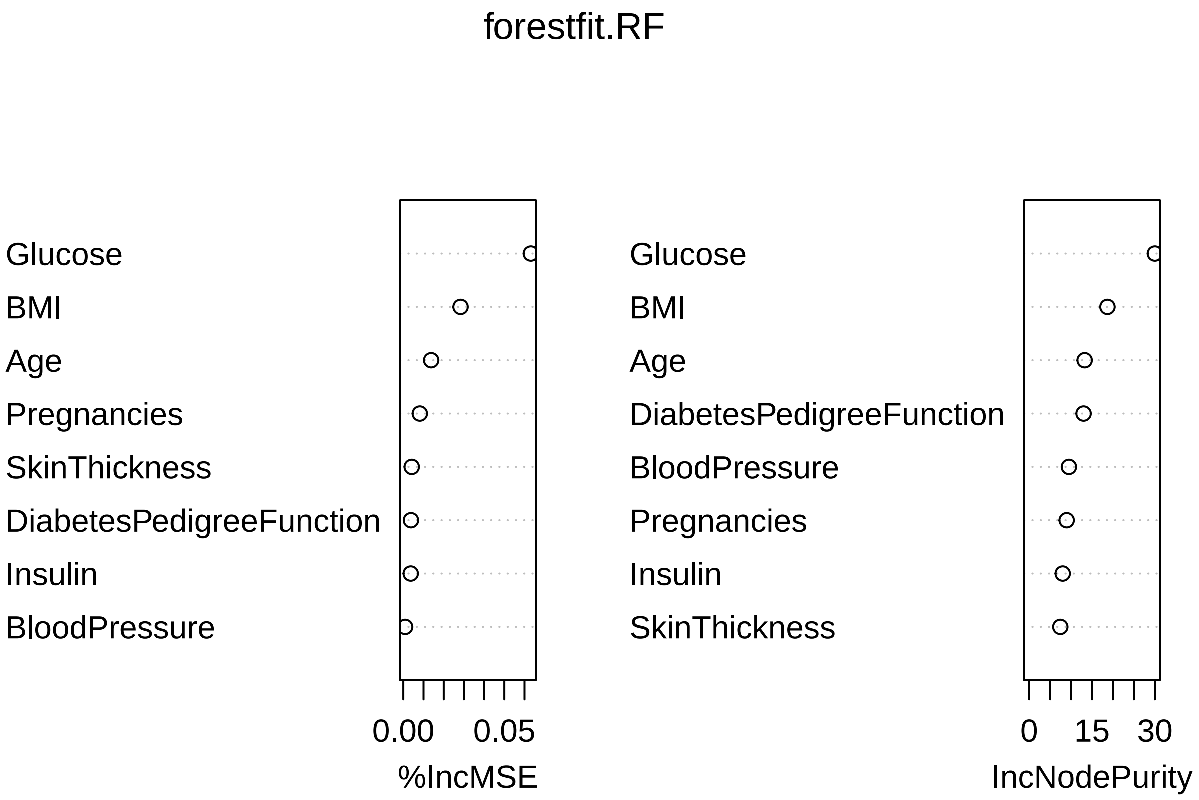
```
##
## Attaching package: ’randomForest’
```

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

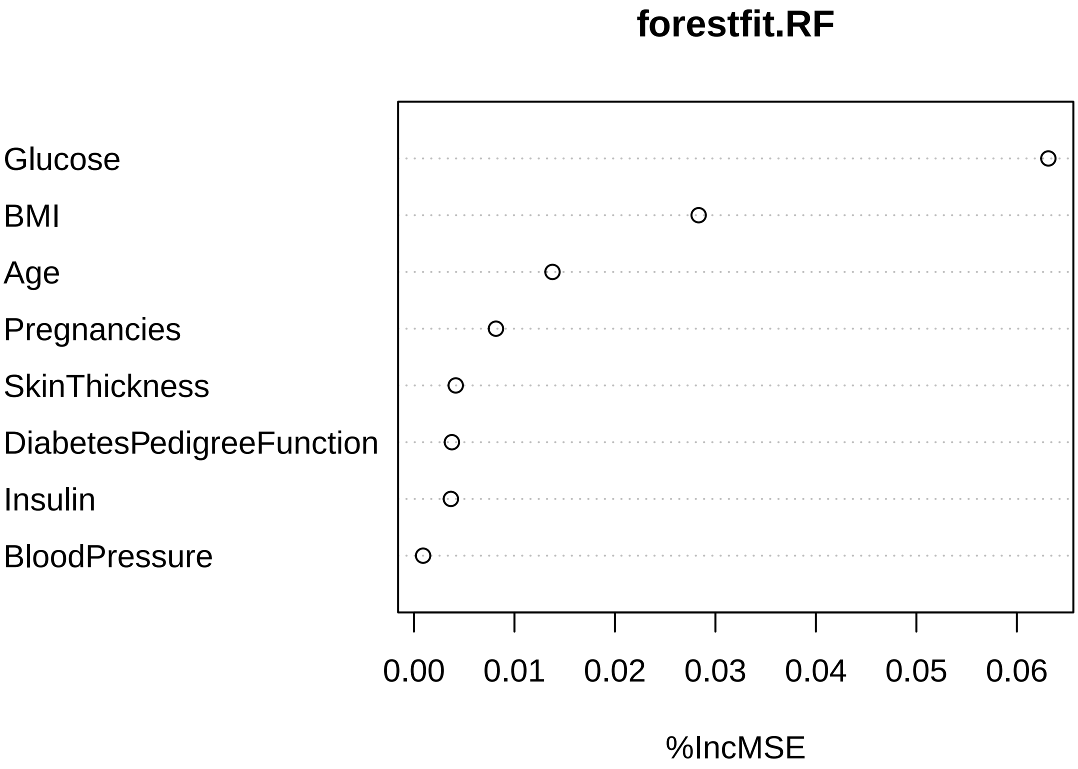
```
forestfit.RF <- randomForest(Outcome ~., data = bn.training, ntree = 500, importance = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

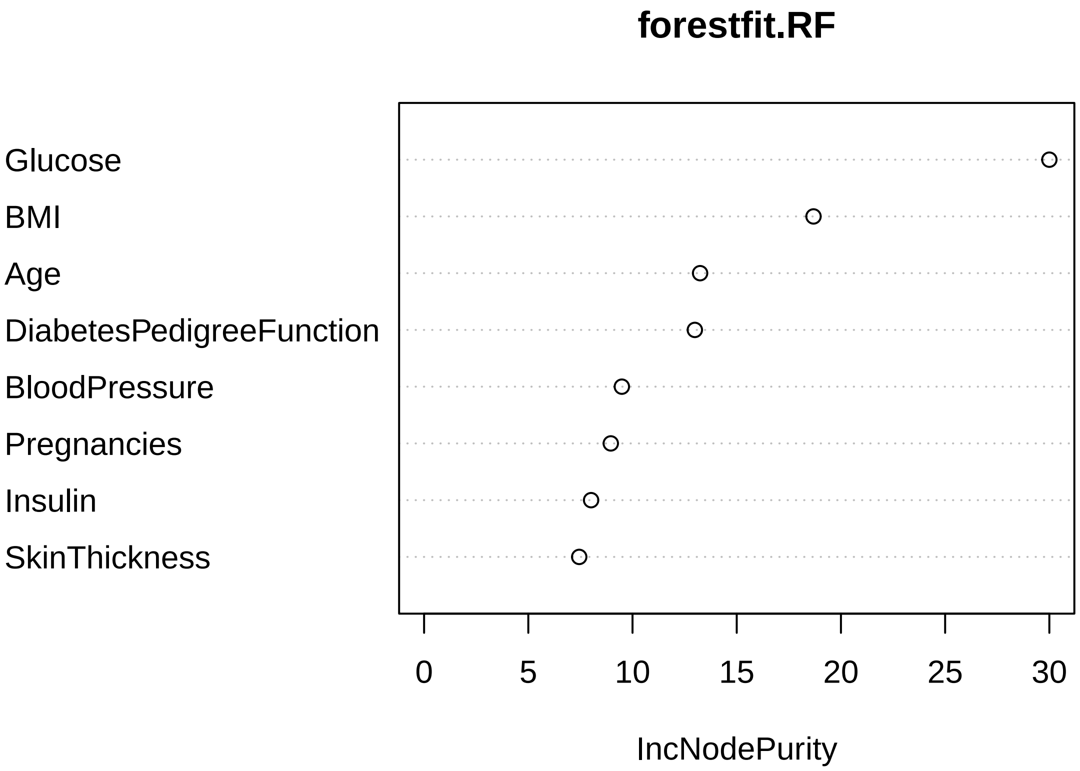
```
varImpPlot(forestfit.RF, scale = F)
```



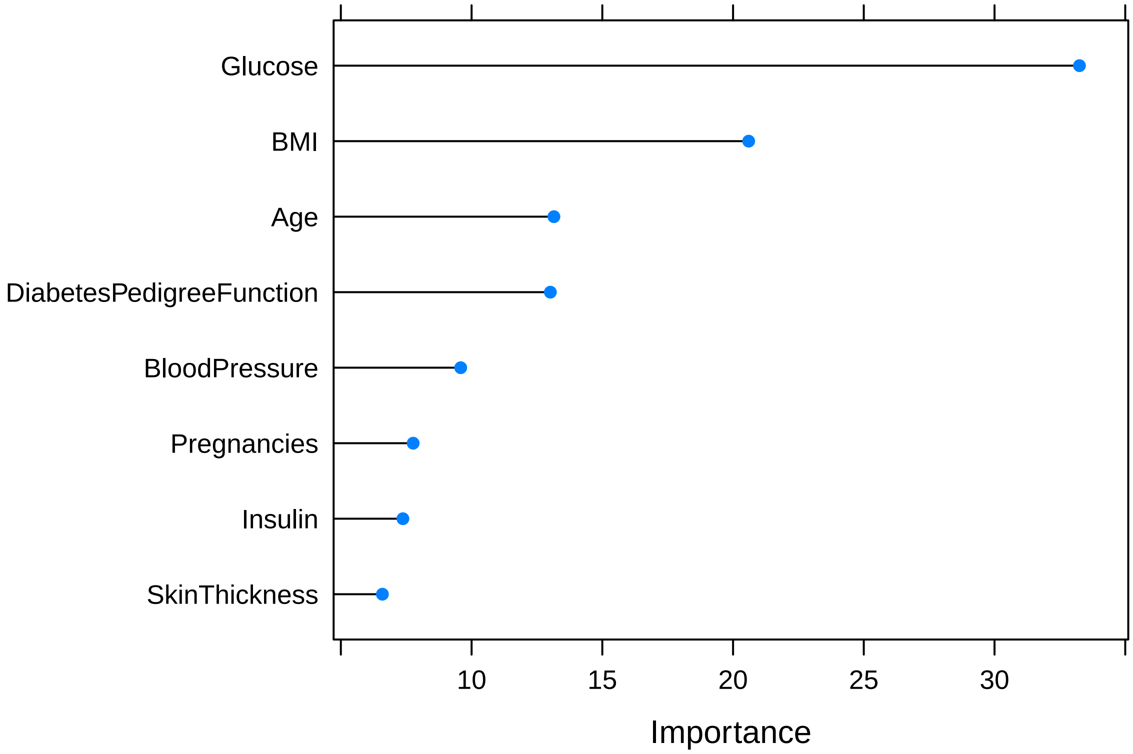
```r
varImpPlot(forestfit.RF, type = 1, scale = F)
```



```r
varImpPlot(forestfit.RF, type = 2, scale = F)
```



```
RFimp <- varImp(forestfit.m, scale = F)
plot(RFimp)
```



According to the result above, "Glucose" is the most important variable.