Adv. Stat. Topics A

Day 3

Lectures about trees and forests

Anne Lyngholm Sørensen (als@sund.ku.dk)

Thomas Alexander Gerds (tag@biostat.ku.dk)

Helene Charlotte Wiese Rytgaard (hely@sund.ku.dk)

Section of Biostatistics, Department of Public Health

Outline

- Modeling cultures
- Model selection
- Decision trees
- From trees to forests
- Tuning random forests
- Variable importance

Software Overview (see also `http://philipppro.github.io/More_complete_list/`)

	Outcome				
Package	Continuous	Binary	Survival	Comp. risks	Method
rpart[1]	X	X	X		Tree
randomForest[2]	X	X			Forest
party[3]	X	X	X		Tree/Forest
randomForestSRC[4]	X	X	X	X	Forest
ranger[5]	X	X	X		Forest

[1] `rpart` Therneau, Atkinson and Ripley
[2] `randomForest` Liaw and Wiener (based on Breiman and Cutler)
[3] `ctree, cforest` Hothorn
[4] `rfsrc` Ishwaran
[5] `ranger` Wright and Ziegler

Targets of analysis: Random forests are used to

- predict individual outcome
- rank and select variables

in particular in high dimensional settings where the number of variables exceeds the number of subjects in the dataset.

Targets of analysis: Random forests are used to

- predict individual outcome
- rank and select variables

in particular in high dimensional settings where the number of variables exceeds the number of subjects in the dataset.

Example: binary outcome Y

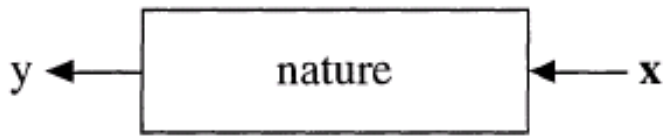
$$Y = \begin{cases} 1 & \text{event} \\ 0 & \text{no event} \end{cases}$$

Type	Prediction	Error
Class	c = either 0 or 1	Y==c
Probability	p= value between 0 and 1	$(Y - p)^2$

Random forest can be an alternative to logistic regression

The two cultures

Statistics starts with data. Think of the data as being generated by a black box in which a vector of input variables **x** (independent variables) go in one side, and on the other side the response variables **y** come out. Inside the black box, nature functions to associate the predictor variables with the response variables, so the picture is like this:



There are two goals in analyzing the data:

Prediction. To be able to predict what the responses are going to be to future input variables;
Information. To extract some information about how nature is associating the response variables to the input variables.

L. Breiman. Statistical modeling: The two cultures. Statistical Science, 16 (3):199-215, 2001.

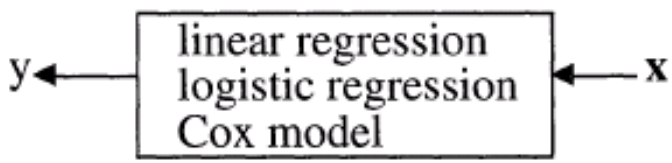
The two cultures

The Data Modeling Culture

The analysis in this culture starts with assuming a stochastic data model for the inside of the black box. For example, a common data model is that data are generated by independent draws from

response variables = f(predictor variables, random noise, parameters)

The values of the parameters are estimated from the data and the model then used for information and/or prediction. Thus the black box is filled in like this:

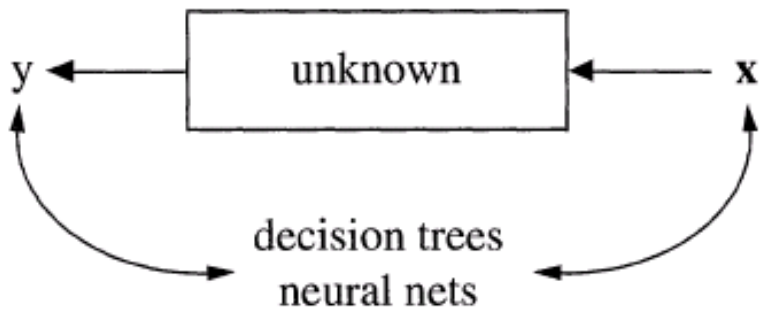


Model validation. Yes–no using goodness-of-fit tests and residual examination.
Estimated culture population. 98% of all statisticians.

The two cultures

The Algorithmic Modeling Culture

The analysis in this culture considers the inside of the box complex and unknown. Their approach is to find a function $f(\mathbf{x})$—an algorithm that operates on $\mathbf{x}$ to predict the responses $\mathbf{y}$. Their black box looks like this:



Model validation. Measured by predictive accuracy.
Estimated culture population. 2% of statisticians, many in other fields.

Example: epo study

Anaemia is a deficiency of red blood cells and/or hemoglobin and an additional risk factor for cancer patients.

Randomized placebo controlled trial[6]: does treatment with epoetin beta – epo – (300 U/kg) enhance hemoglobin concentration level and improve survival chances?

Henke et al. 2006 identified the c20 expression (erythropoietin receptor status) as a new biomarker for the prognosis of locoregional progression-free survival.

[6] Henke et al. Do erythropoietin receptors on cancer cells explain unexpected clinical findings? J Clin Oncol, 24(29):4708-4713, 2006.

Treatment

The study includes 149 [7]head and neck cancer patients with a tumor located in the oropharynx (36%), the oral cavity (27%), the larynx (14%) or in the hypopharynx (23%).

One of the treatments was radiotherapy following

	Resection		
	Complete	Incomplete	No
Placebo	35	14	25
Epo	36	14	25

[7]with non-missing blood values

Outcome

Blood hemoglobin levels were measured weekly during radiotherapy (7 weeks).

Treatment with epoetin beta was defined successful when the hemoglobin level increased sufficiently. For patient i set

$$Y_i = \begin{cases} 1 & \text{treatment successful} \\ 0 & \text{treatment failed} \end{cases}$$

Target

Patient no.	Treatment successful	Predicted probability
1	0	P_1
2	0	P_2
3	1	P_3
4	1	P_4
5	0	P_5
6	1	P_6
7	1	P_7
.	.	.
.	.	.

Predictors

Age	min: 41 y, median: 59 y, max: 80 y
Gender	male: 85%, female: 15%
Baseline hemoglobin	mean: 12.03 g/dl, std: 1.45
Treatment	epo: 50%, placebo 50%
Resection	complete: 48%, incomplete: 19%, no resection: 34%
Epo receptor status	neg: 32%, pos: 68%

Logistic regression

Response: treatment successful yes/no

Factor	OddsRatio	StandardError	CI.95	pValue
(Intercept)	0.00	4.01	–	< 0.0001
Age	0.97	0.03	$[0.91; 1.03]$	0.2807
Sex:female	4.71	0.84	$[0.91; 26.02]$	0.0657
HbBase	3.25	0.27	$[1.99; 5.91]$	< 0.0001
Treatment:Epo	90.92	0.76	$[23.9; 493.41]$	< 0.0001
Resection:Incompl	1.75	0.81	$[0.36; 9.03]$	0.4924
Resection:Compl	4.14	0.69	$[1.13; 17.36]$	0.0395
Receptor:positive	5.81	0.66	$[1.72; 23.39]$	0.0076

Logistic regression

Response: treatment successful yes/no

Factor	OddsRatio	StandardError	CI.95	pValue
(Intercept)	0.00	4.01	–	< 0.0001
Age	0.97	0.03	$[0.91; 1.03]$	0.2807
Sex:female	4.71	0.84	$[0.91; 26.02]$	0.0657
HbBase	3.25	0.27	$[1.99; 5.91]$	< 0.0001
Treatment:Epo	90.92	0.76	$[23.9; 493.41]$	< 0.0001
Resection:Incompl	1.75	0.81	$[0.36; 9.03]$	0.4924
Resection:Compl	4.14	0.69	$[1.13; 17.36]$	0.0395
Receptor:positive	5.81	0.66	$[1.72; 23.39]$	0.0076

Does that mean everyone should be treated?

The model provides information for a single patient

For example: the predicted probability that a 51 year old man with complete tumor resection and baseline hemoglobin level 12.6 g/dl reaches the target hemoglobin level (Y_i=1) is

Epo treatment: 97.4%

Placebo group: 29.2 %

The model provides information for a single patient

For example: the predicted probability that a 51 year old man with complete tumor resection and baseline hemoglobin level 12.6 g/dl reaches the target hemoglobin level (Y_i=1) is

Epo treatment: 97.4%

Placebo group: 29.2 %

If a similar patient has baseline hemoglobin level 14.8 g/dl then the model predicts:

Epo treatment: 99.8%

Placebo group: 84.7 %

Model selection

Very many different 'logistic regression models' can be constructed by selecting subsets of variables, transformations, and interactions of variables.

"Standard" multiple (logistic) regression works if

- the number of predictors is not too large, and substantially smaller than the sample size
- the decision maker has a-priory knowledge about which variables to put into the model

Ad-hoc model selection algorithms, like automated backward elimination, do not lead to reproducible prediction models.

ELSEVIER

Journal of Clinical Epidemiology 57 (2004) 1138–1146

Journal of Clinical Epidemiology

Automated variable selection methods for logistic regression produced unstable models for predicting acute myocardial infarction mortality

Peter C. Austin[a,b,c,*], Jack V. Tu[a,b,c,d,e]

Abstract

Objectives: Automated variable selection methods are frequently used to determine the independent predictors of an outcome. T objective of this study was to determine the reproducibility of logistic regression models developed using automated variable selecti methods.

Study Design and Setting: An initial set of 29 candidate variables were considered for predicting mortality after acute myocard infarction (AMI). We drew 1,000 bootstrap samples from a dataset consisting of 4,911 patients admitted to hospital with an AMI. Usi each bootstrap sample, logistic regression models predicting 30-day mortality were obtained using backward elimination, forward selecti and stepwise selection. The agreement between the different model selection methods and the agreement across the 1,000 bootstrap samp were compared.

Results: Using 1,000 bootstrap samples, backward elimination identified 940 unique models for predicting mortality. Similar resu were obtained for forward and stepwise selection. Three variables were identified as independent predictors of mortality among all bootstr samples. Over half the candidate prognostic variables were identified as independent predictors in less than half of the bootstrap sampl

Conclusion: Automated variable selection methods result in models that are unstable and not reproducible. The variables selected independent predictors are sensitive to random fluctuations in the data.

Keywords: Regression models; Multivariate analysis; Variable selection; Logistic regression; Acute myocardial infarction; Epidemiology

Backward elimination

On full data (n=149):

```
library(rms)
full <- lrm(Y~age+sex+HbBase+Treat+Resection+Receptor,data=Epo)
fastbw(full)
bw <- lrm(Y~sex+HbBase+Treat+Receptor,data=Epo)
```

```
 Deleted    Chi-Sq d.f. P      Residual d.f. P      AIC
 age        1.16   1    0.2807 1.16     1    0.2807 -0.84
 Resection 3.75   2    0.1532 4.92     3    0.1781 -1.08

Approximate Estimates after Deleting Factors

                     Coef   S.E. Wald Z             P
Intercept         -11.257 3.0129 -3.736 0.00018665428
sex=male           -1.672 0.8221 -2.034 0.04195853231
HbBase              1.099 0.2719  4.043 0.00005279348
Treat=Placebo      -3.843 0.6992 -5.496 0.00000003887
Receptor=positive   1.413 0.6355  2.224 0.02615849462

Factors in Final Model

[1] sex      HbBase   Treat    Receptor
```

Backward elimination

On reduced data (n=130):

```
library(rms)
set.seed(17)
Epo17 <- Epo[sample(1:149,replace=FALSE,size=130),]
sub <- lrm(Y~age+sex+HbBase+Treat+Resection+Receptor,
       data=Epo17)
fastbw(sub)
subbw <- lrm(Y~HbBase+Treat, data=Epo17)
```

```
 Deleted    Chi-Sq d.f. P      Residual d.f. P      AIC
 age        0.61   1    0.4362 0.61     1    0.4362 -1.39
 Resection 4.81    2    0.0905 5.41     3    0.1440 -0.59

Approximate Estimates after Deleting Factors

                     Coef   S.E. Wald Z           P
Intercept         -11.657 3.2936 -3.539 0.000401291
sex=male           -1.692 0.8643 -1.958 0.050277808
HbBase              1.127 0.2954  3.816 0.000135846
Treat=Placebo      -3.370 0.6971 -4.833 0.000001343
Receptor=positive   1.311 0.6639  1.974 0.048333452

Factors in Final Model

[1] sex      HbBase   Treat    Receptor
```

Predicted chance of treatment success for a new patient

```
newpatient
```

```
  age  sex HbBase Treat Resection Receptor
1  48 male   10.8   Epo        No negative
```

```
pfull=predictRisk(full,newdata=newpatient)
pbw=predictRisk(bw,newdata=newpatient)
psubbw=predictRisk(subbw,newdata=newpatient)
# table results
res=cbind(round(100*c(pfull,pbw,psubbw),1))
rownames(res)=c("Full model","BW all data","BW subset")
colnames(res)=c("Predicted chance (%)")
res
```

```
             Predicted chance (%)
Full model                   16.9
BW all data                  24.1
BW subset                    47.8
```

Exercise

Load the Epo data:

```
Epo <- read.csv("http://publicifsv.sund.ku.dk/~helene/
    Epo.csv", stringsAsFactors=TRUE)
```

Epo data set is ready for analysis

- Choose your favorite seed to generate a subsample (n=130) of the Epo data
- Run backward elimination with function rms::fastbw
- Predict the outcome for the following new patient

```
newpatient <- read.csv("http://publicifsv.sund.ku.dk/~
    helene/newpatient", stringsAsFactors=TRUE)
```

- Report the selected variables and the predicted risk

Decision trees

A Conversation of Richard Olshen with Leo Breiman

. . .

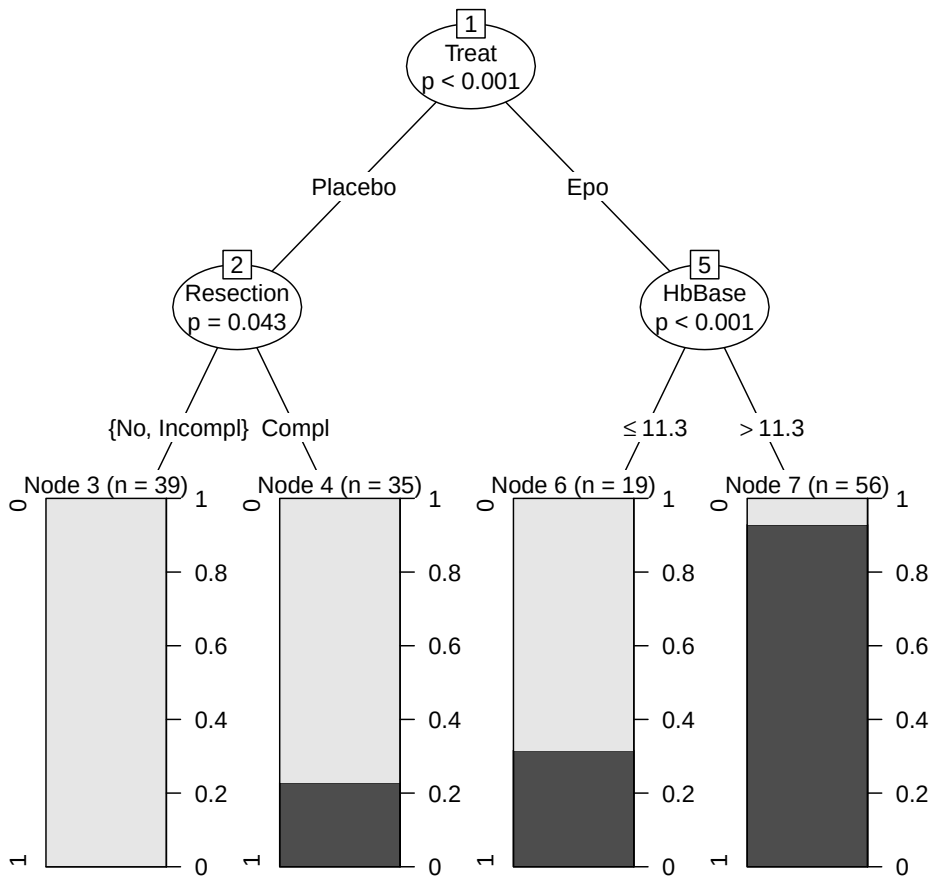
Olshen: *What about arcing, bagging and boosting?*
Breiman: *Okay. Yeah. This is fascinating stuff, Richard. In the last five years, there have been some really big breakthroughs in prediction. And I think combining predictors is one of the two big breakthroughs. And the idea of this was, okay, that suppose you take CART, which is a pretty good classifier, but not a great classifier. I mean, for instance, neural nets do a much better job.*
Olshen: *Well, suitably trained?*
Breiman: Suitably trained.
Olshen: *Against an untrained CART?*

Breiman: *Right. Exactly. And I think I was thinking about this. I had written an article on subset selection in linear regression. I had realized then that* **subset selection in linear regression is really a very unstable procedure**. *If you tamper with the data just a little bit, the first best five variable regression may change to another set of five variables. And so I thought, "Okay. We can stabilize this by just perturbing the data a little and get the best five variable predictor. Perturb it again. Get the best five variable predictor and then average all these five variable predictors." And sure enough, that worked out beautifully. This was published in an article in the Annals (Breiman, 1996b).*

. . .

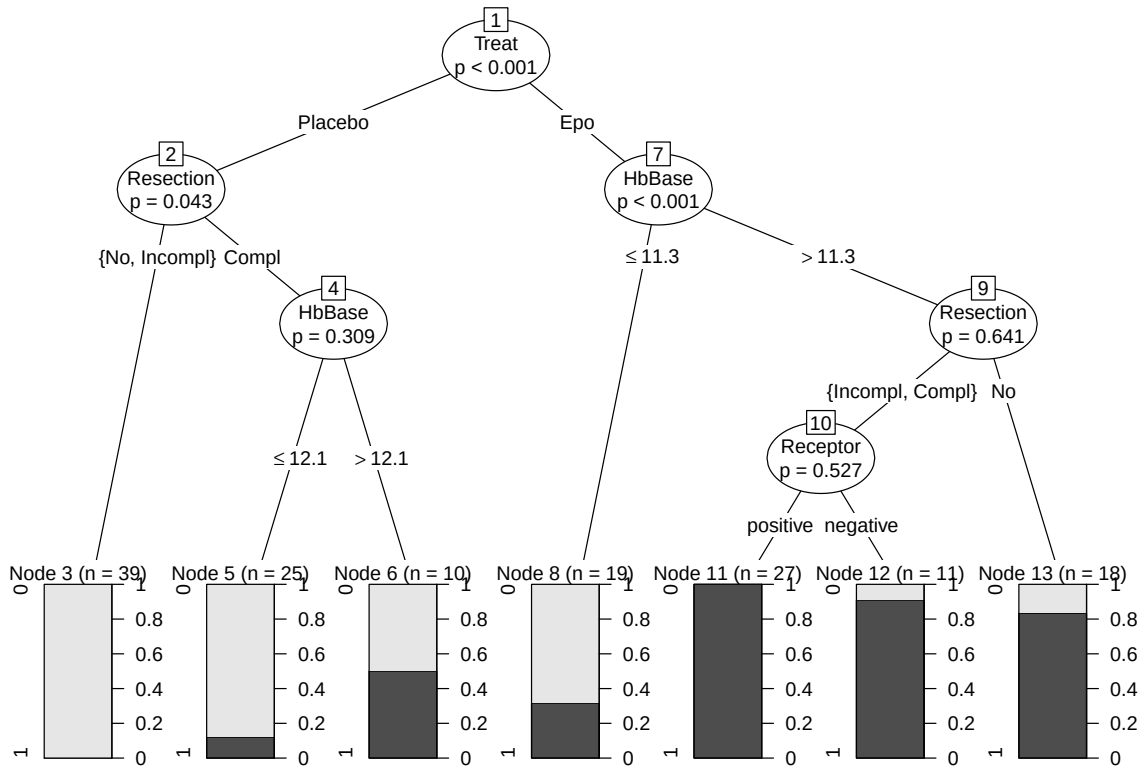
Conditional inference trees are not very deep (by default)

```
library(party)
plot(ctree(Y~age+sex+HbBase+Treat+Resection+Receptor,data=
    Epo))
```



A deeper more greedy tree

```
library(party)
plot(ctree(Y~age+sex+HbBase+Treat+Resection+Receptor,data=
    Epo,controls=ctree_control(mincriterion = .01)))
```



Classification trees

A tree model is a form of recursive partitioning.

It lets the data decide which variables are important and where to place cut-offs in continuous variables.

In general terms, the purpose of the analyzes via tree-building algorithms is to determine a set of splits that permit accurate prediction or classification of cases.

In other words: a tree is a combination of many medical tests.

Roughly, the algorithm works as follows:

1. Find the predictor so that the best possible split on that predictor optimizes some statistical criterion over all possible splits on the other predictors.
2. For ordinal and continuous predictors, the split is of the form $X < c$ versus $X \geq c$.
3. Repeat step 1 within each previously formed subset.
4. Proceed until fewer than k observations remain to be split, or until nothing is gained from further splitting, i.e. the tree is fully grown.
5. The tree is pruned according to some criterion.

Characters of classification trees

- Trees are specifically designed for accurate classification/prediction
- Results have a graphical representation and are easy to interpret
- No model assumptions
- Recursive partitioning can identify complex interactions
- One can introduce different costs of miss-classification in the tree

But:

- Trees are not robust against even small perturbations of the data.
- It is quite easy to over-fit the data.
- Trees are weak learners

Random forests

Growing trees into forests

Decision trees are nice because:

- They produce results that are easy to interpret
- They require no model assumptions

But:

- They easily overfit the data
- Trees are weak learners

Random forests can overcome some of these weaknesses by combining multiple decision trees.

Outline

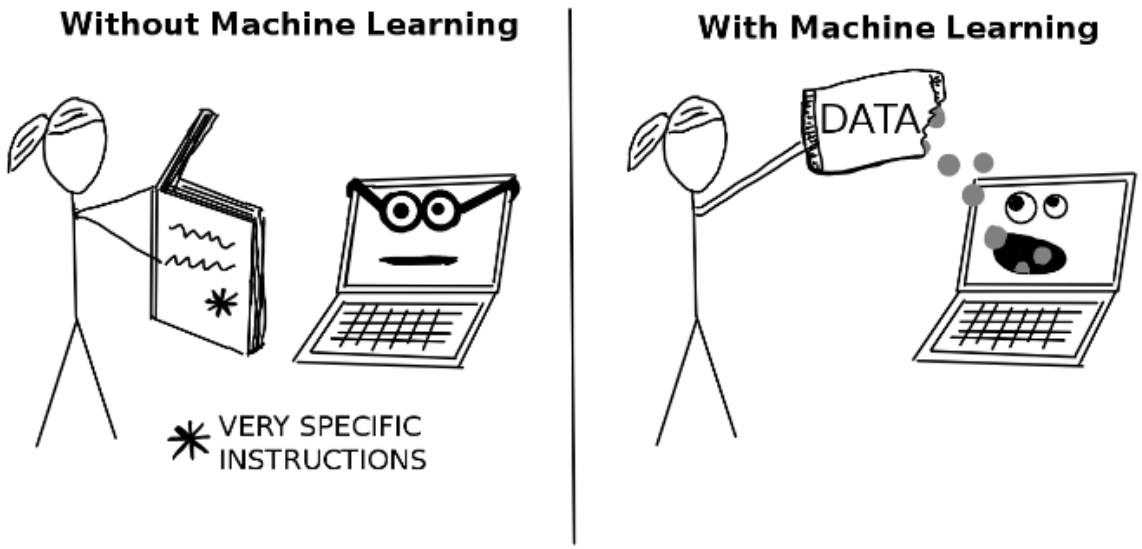
- ~~Modeling cultures~~
- ~~Model selection~~
- ~~Decision trees~~
- From trees to forests
- Tuning random forests
- Variable importance

Outline

- ~~Modeling cultures~~
- ~~Model selection~~
- ~~Decision trees~~
- From trees to forests
  - Random Forest and Machine Learning
  - Bootstrapping
  - Random subspace
  - Ensemble methods
- Tuning random forests
- Variable importance

Random forests and machine learning

Random forests is a machine learning method.



When is Random Forests cool?

It is cool when:

- You want to predict an outcome where prediction accuracy is key.
- You want a machine to help in selecting what is important for getting better predictions.
  - You do not know the underlying dynamics of the outcome
  - You do not have a specific hypothesis

Genomics 99 (2012) 323–329

Contents lists available at SciVerse ScienceDirect

Genomics

journal homepage: www.elsevier.com/locate/ygeno

Review

Random forests for genomic data analysis

Xi Chen *, Hemant Ishwaran

Department of Biostatistics, Vanderbilt University, Nashville, TN 37232, USA
Division of Biostatistics, Department of Epidemiology and Public Health, University of Miami, Miami, FL 33136, USA

ARTICLE INFO

Article history:
Received 9 February 2012
Accepted 14 April 2012
Available online 21 April 2012

Keywords:
Random forests
Random survival forests
Classification
Prediction
Variable selection
Genomic data analysis

ABSTRACT

Random forests (RF) is a popular tree-based ensemble machine learning tool that is highly data adaptive, applies to "large p, small n" problems, and is able to account for correlation as well as interactions among features. This makes RF particularly appealing for high-dimensional genomic data analysis. In this article, we systematically review the applications and recent progresses of RF for genomic data, including prediction and classification, variable selection, pathway analysis, genetic association and epistasis detection, and unsupervised learning.

Circulation: Cardiovascular Quality and Outcomes
Volume 4, Issue 1, January 2011; Pages 39-45
https://doi.org/10.1161/CIRCOUTCOMES.110.939371

ORIGINAL ARTICLE

Identifying Important Risk Factors for Survival in Patient With Systolic Heart Failure Using Random Survival Forests

Eileen Hsich, MD, Eiran Z. Gorodeski, MD, MPH, Eugene H. Blackstone, MD, Hemant Ishwaran, PhD, and Michael S. Lauer, MD

BACKGROUND— Heart failure survival models typically are constructed using Cox proportional hazards regression. Regression modeling suffers from a number of limitations, including bias introduced by commonly used variable selection methods. We illustrate the value of an intuitive, robust approach to variable selection, random survival forests (RSF), in a large clinical cohort. RSF are a potentially powerful extensions of classification and regression trees, with lower variance and bias.

ORIGINAL ARTICLE: PDF ONLY

The AUGIS Survival Predictor

Prediction of Long-term and Conditional Survival after Esophagectomy Using Random Survival Forests

Rahman, Saqib A. MRCS[*,¶]; Walker, Robert C. MRCS[*]; Maynard, Nick FRCS[†]; Trudgill, Nigel MBBS[‡]; Crosby, Tom FRCP[§]; Cromwell, David A. PhD[¶]; Underwood, Timothy J. PhD[*] on behalf of the NOGCA project team AUGIS

Author Information

Annals of Surgery: February 17, 2021 - Volume - Issue -
doi: 10.1097/SLA.0000000000004794

OPEN SDC PAP

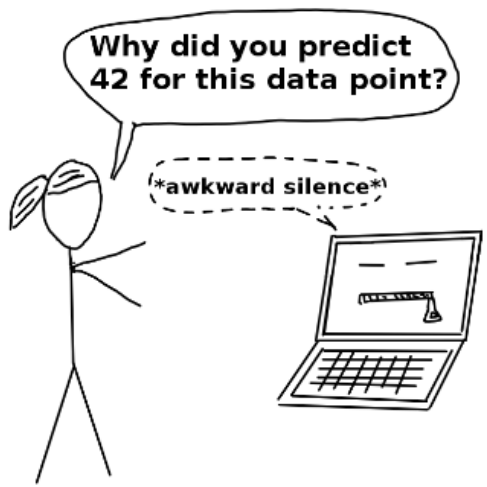
Metrics

When is it then uncool?

You pay by losing some of the traditional statistical inference

This includes how covariates influence the outcome (e.g. usual coefficients from e.g. linear regression models, or summary statistics such as OR, RR and HR).

Hypotheses about causality becomes untestable in the traditional statistical sense.



Easy peasy predictions

A little glimpse into the future:

```
library(randomForestSRC) # load the R library
```

Fitting a random forest is then fairly easy,

```
RFmodel <-
    rfsrc(Y~age+sex+HbBase+Treat+Resection+Receptor,
      data = Epo, mtry = 2, ntree = 500,
      nodesize = 5)
```

And extracting predictions:

```
round(RFmodel$predicted[1:5],3)
```

```
[1] 0.075 0.066 0.956 0.916 0.052
```

Understanding how it works

The primary interest in this course, is that you understand how Random Forests work and some of the considerations to do when fitting a random forest.

The elements

We will go through some of the key-elements of the Random Forests method via the Epo example.

It will aid in understanding specifically four concepts of Random Forests:

1. ~~Decision Trees~~
2. Bootstrapping
3. Random subspace
4. Ensemble methods

Reality vs. simulation

We will use the epo data for the purpose of illustrating, but we will also simulate data. By simulating data, we will know the truth.

Sometimes in statistics, to understand how methods work or perform, we simulate data.

Epo simulation

We take inspiration from the Epo dataset for our simulation. We consider a scenario where we only have access to two predictors:

	age	HbBase
1	70	10.7
2	68	12.7
3	70	13.4
4	55	12.0
5	69	11.2
6	59	13.5

And using those, we want to estimate the probability:

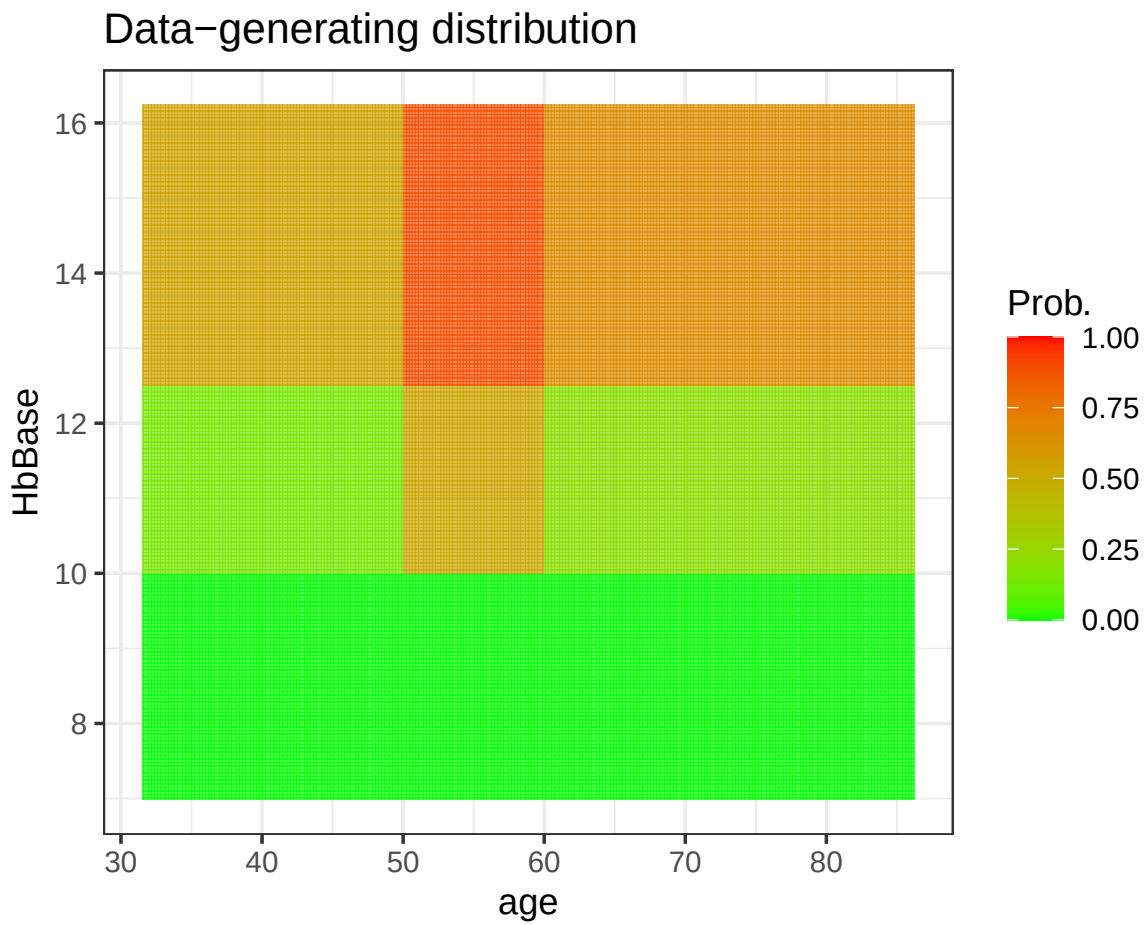
$$P(Y = 1 \mid \text{age}, \text{HbBase})$$

Where $Y = 1$ means treatment is successful.

Epo simulation

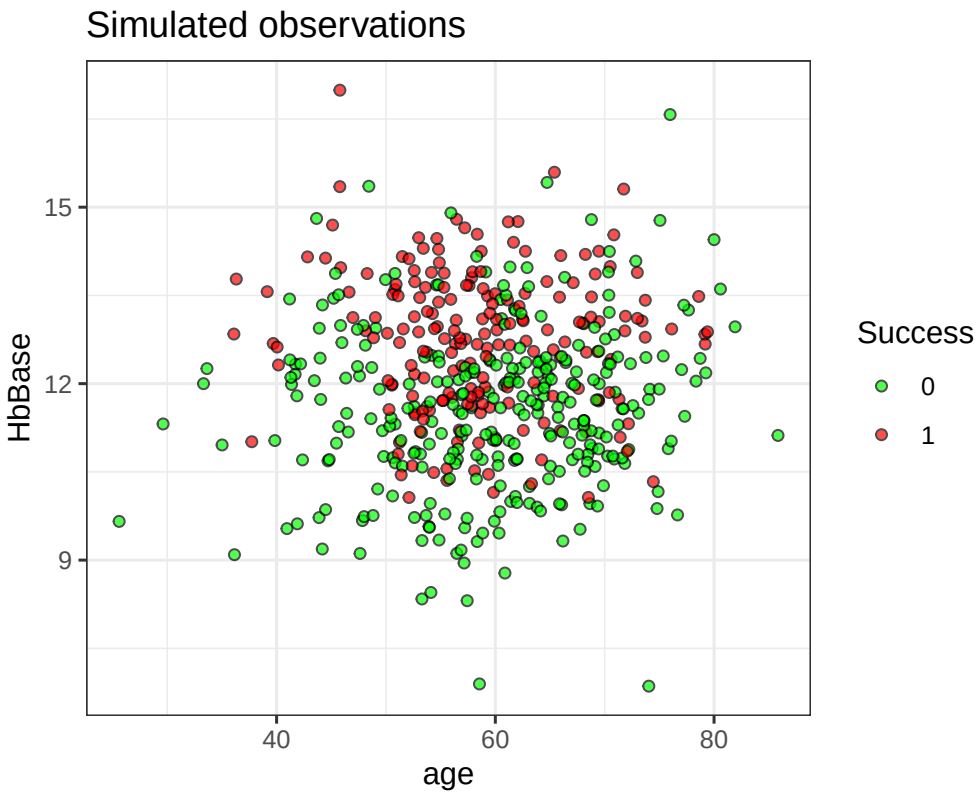
I have simulated data to imitate the Epo data

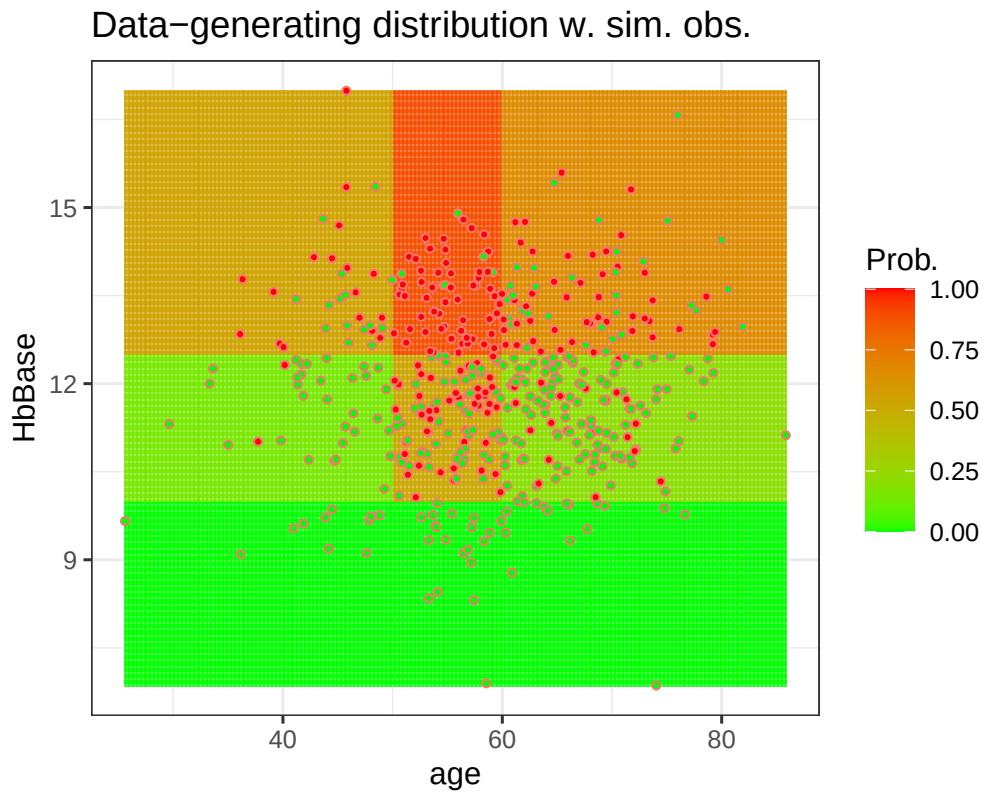
So I know the true $P(Y = 1 \mid \text{age}, \text{HbBase})$



Epo simulation

We can then simulate 500 observations from the data generating distribution.





How to build a Random Forest

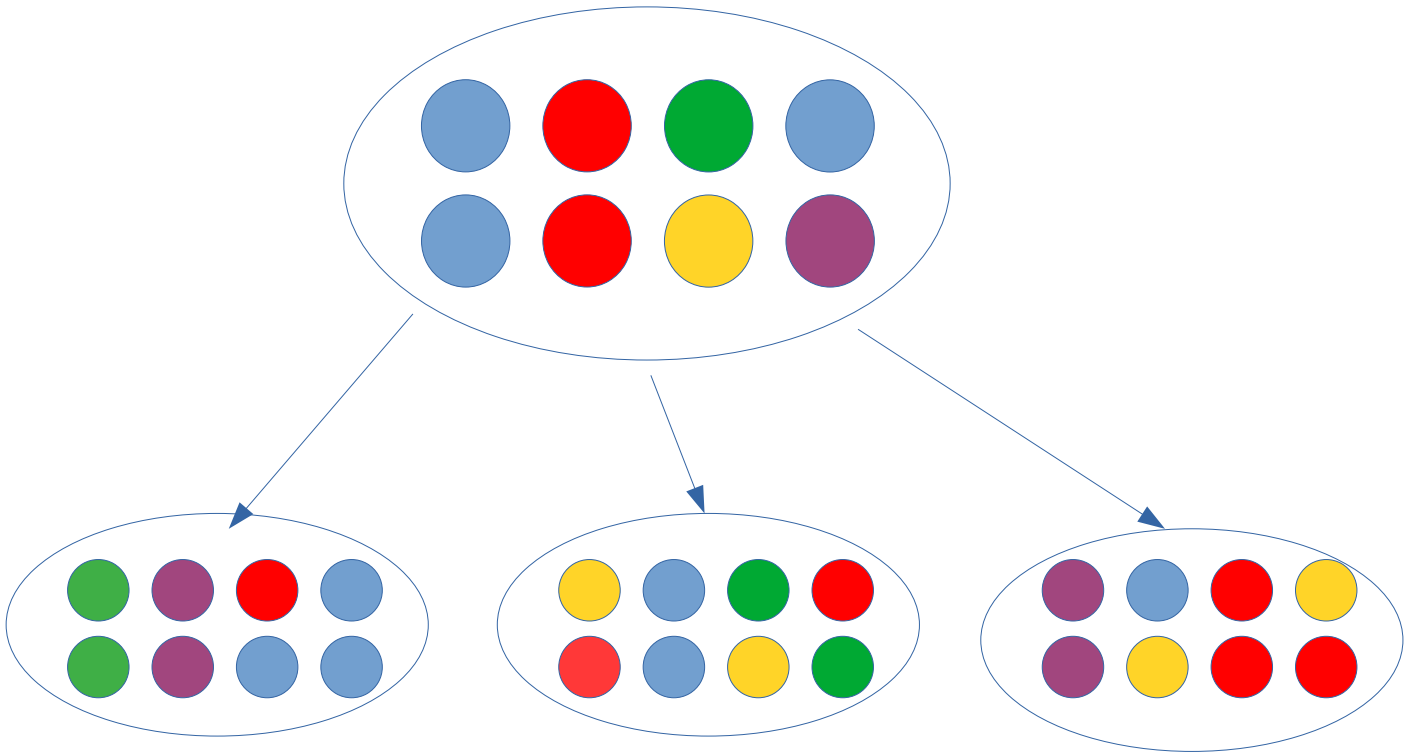
We will explore techniques used inside the forest.

1. Building multiple trees:
   - Using resamples of data (Bootstrapping)
   - Using a subset of the covariates (Random subspace)

2. Averaging over the trees (Ensemble methods)

Resamples of the population

The purpose is creating multiple subpopulations, each of these new populations will be used to fit a decision tree.



Bootstrapping/resampling code

Understanding how bootstrapping works is quite simpe. In R set a seed:

```
set.seed(9)
```

Let n be the number of observations in our data set

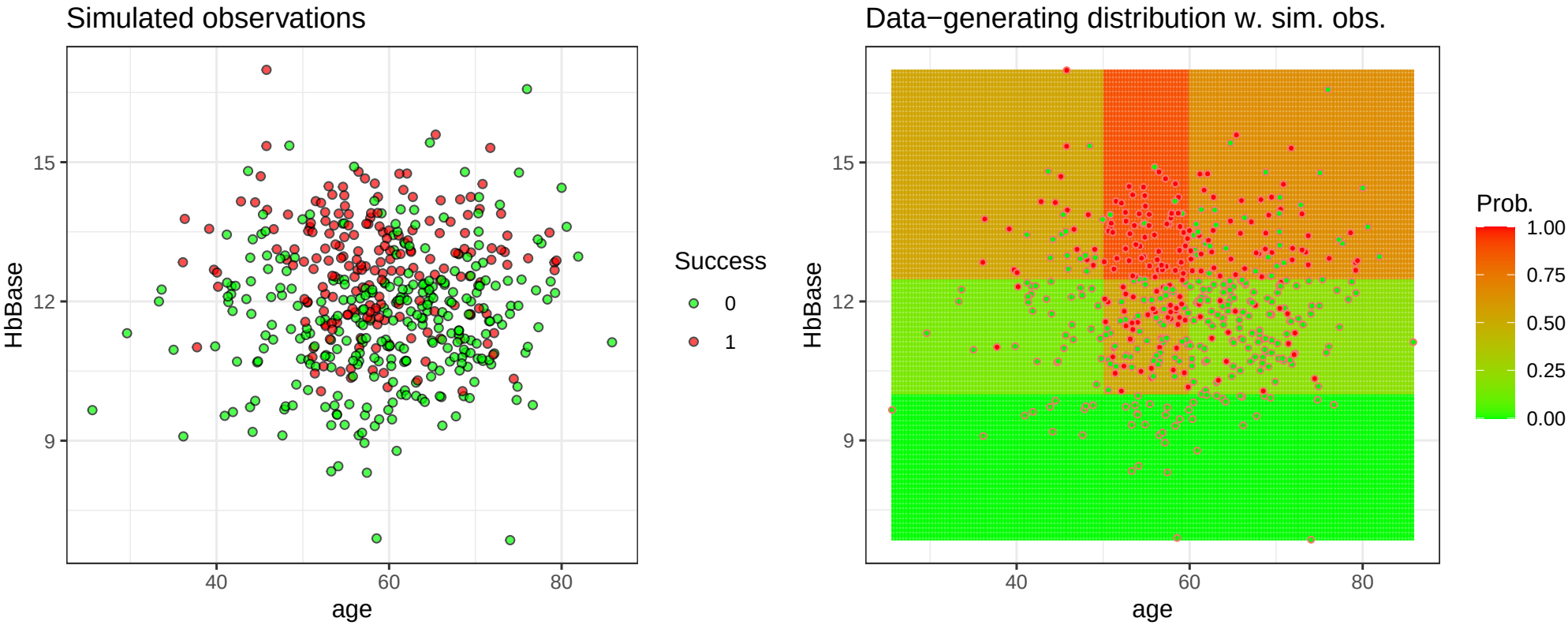
```
n <- 500
bootstrap.sample <- sample(x = 1:n, size = n, replace=
    TRUE)
```

Who is included in the bootstrap sample (look at first six)?

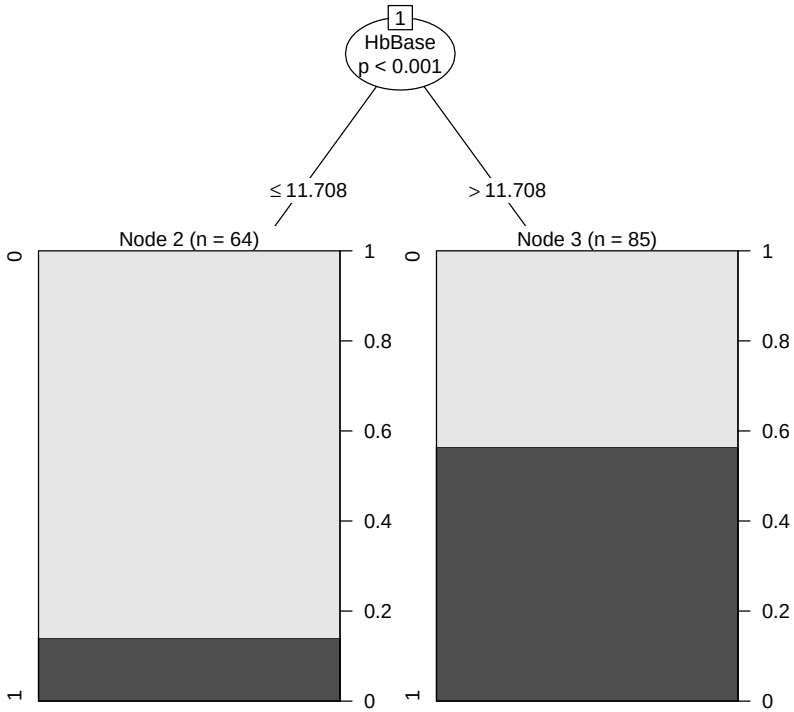
```
head(table(bootstrap.sample))
```

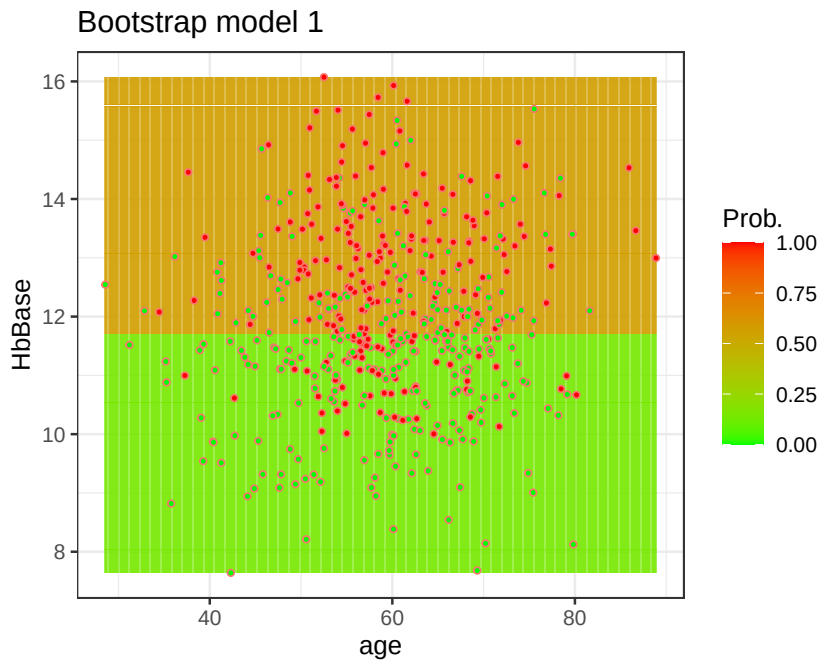
```
bootstrap.sample
1 3 4 5 7 8
1 1 2 1 1 1
```

Epo simulation

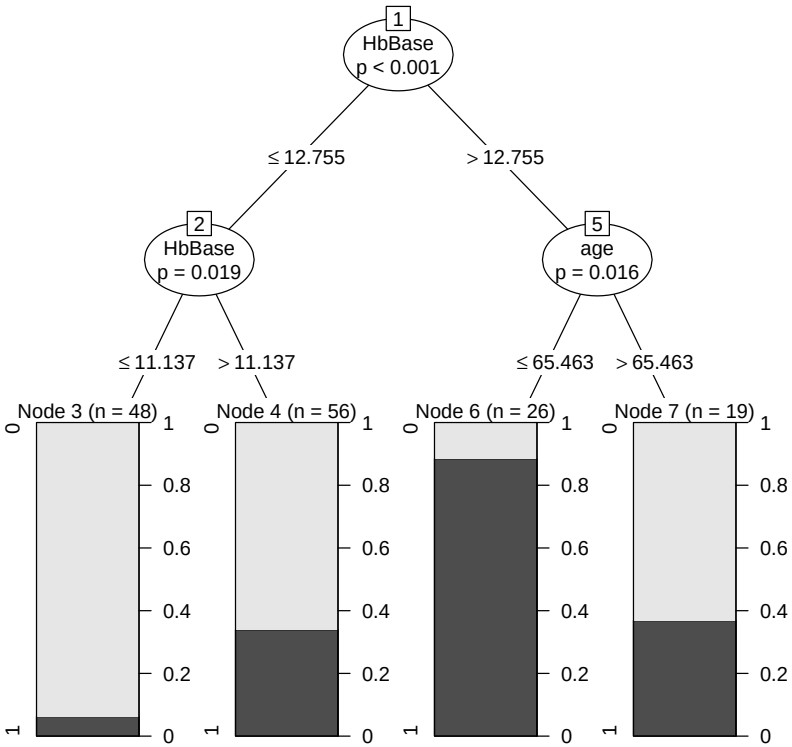


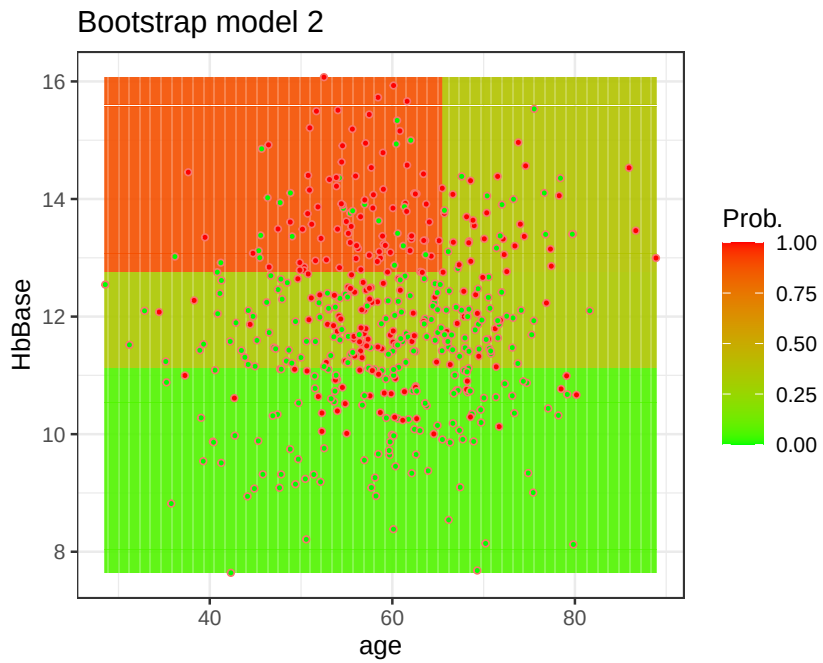
Bootstrap 1 - one tree



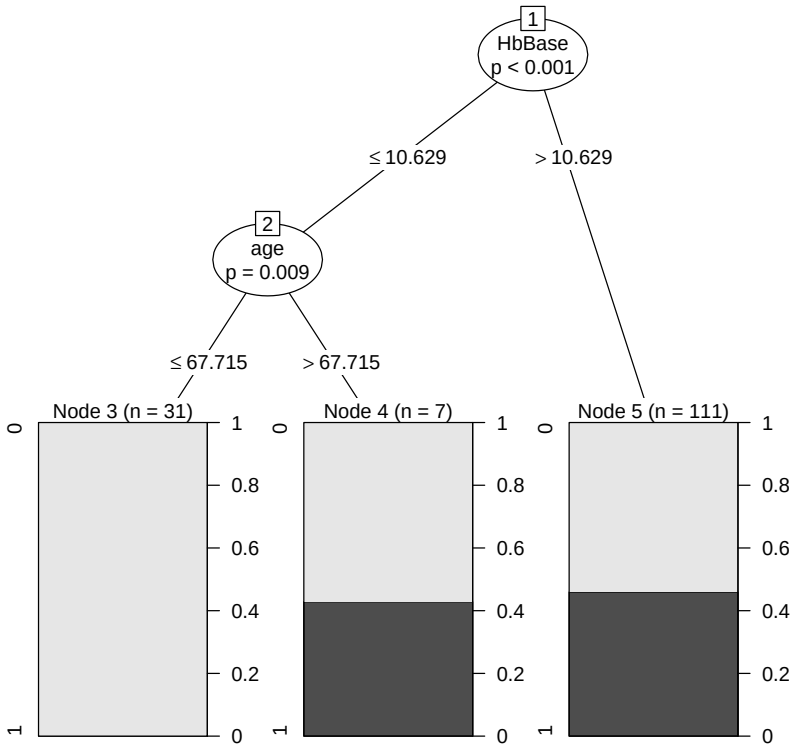


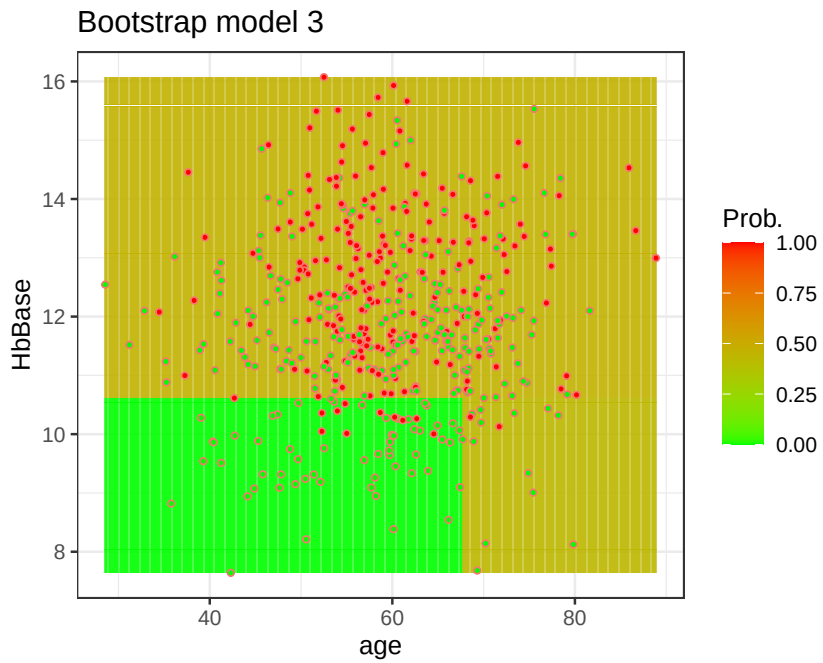
Bootstrap 2 - one tree



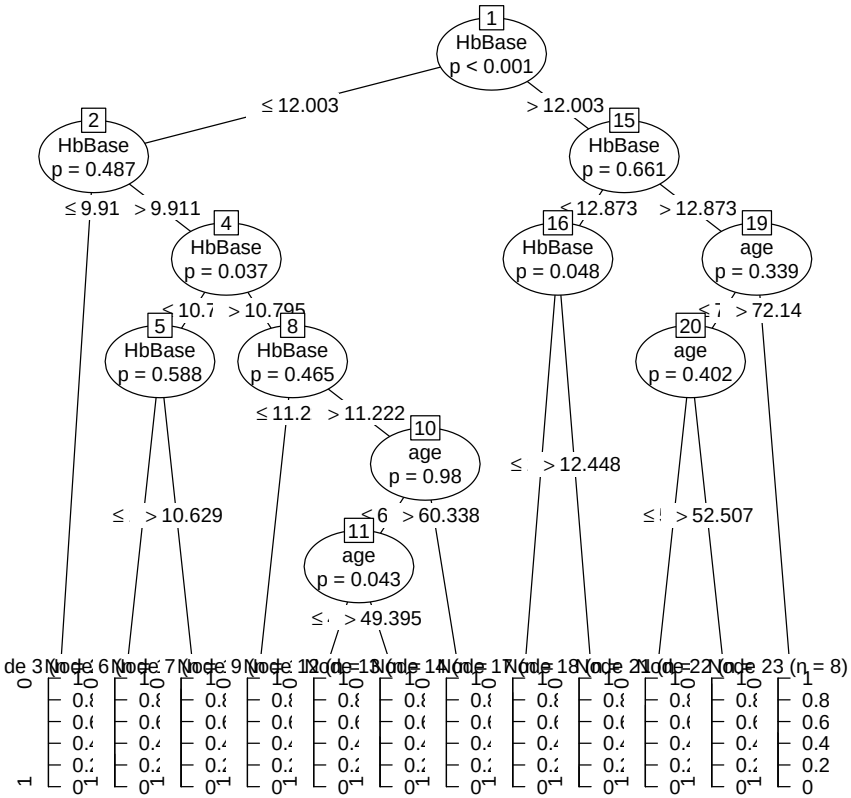


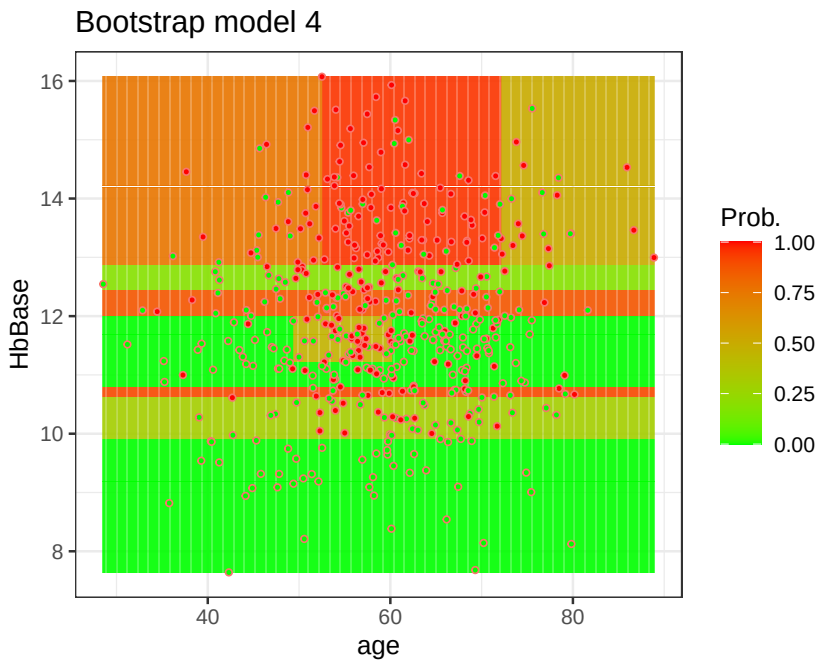
Bootstrap 3 - one tree





Bootstrap 4 (greedy tree) - usually more greedy trees are used in RF





But why? Reducing sensitivity

Random Forests compared to decision trees reduce sensitivity to changes in the data.

We create the model based on present data, but want it to perform well on unseen future data.

If we expect data to be completely the same in the future as in the present, we can use a deep decision tree also as a prediction model. But this assumption is very unusual.

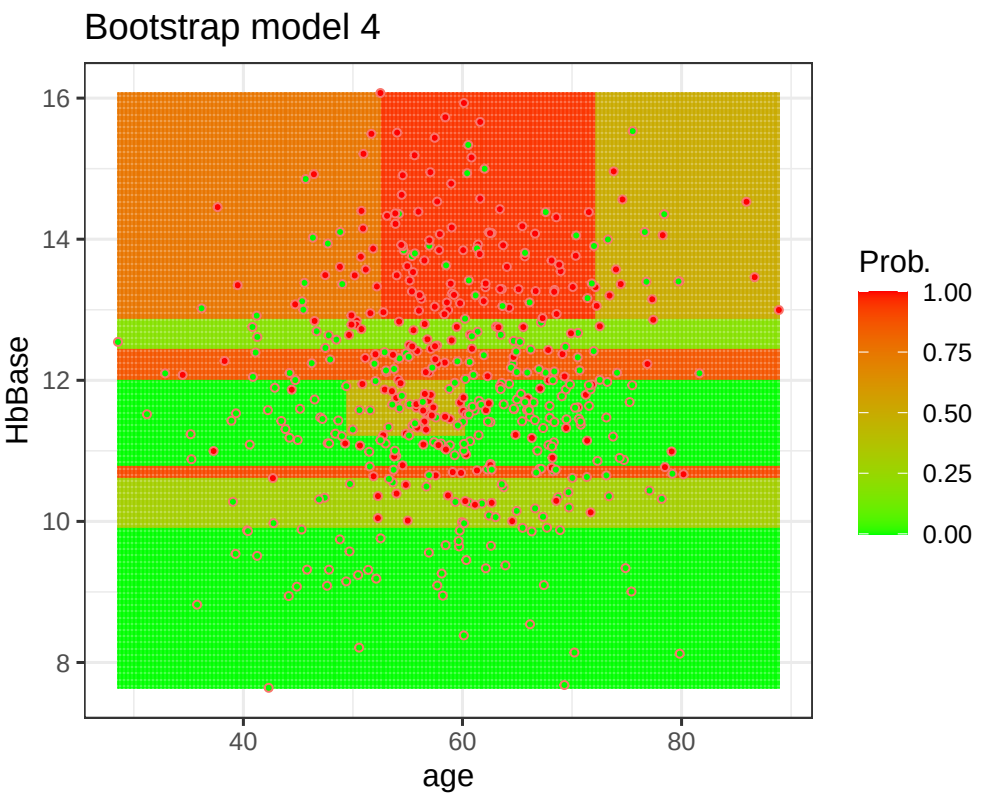
We can try to create versions of the present data, to stabilize the predictions on future data. The versions are created using Bootstrapping.

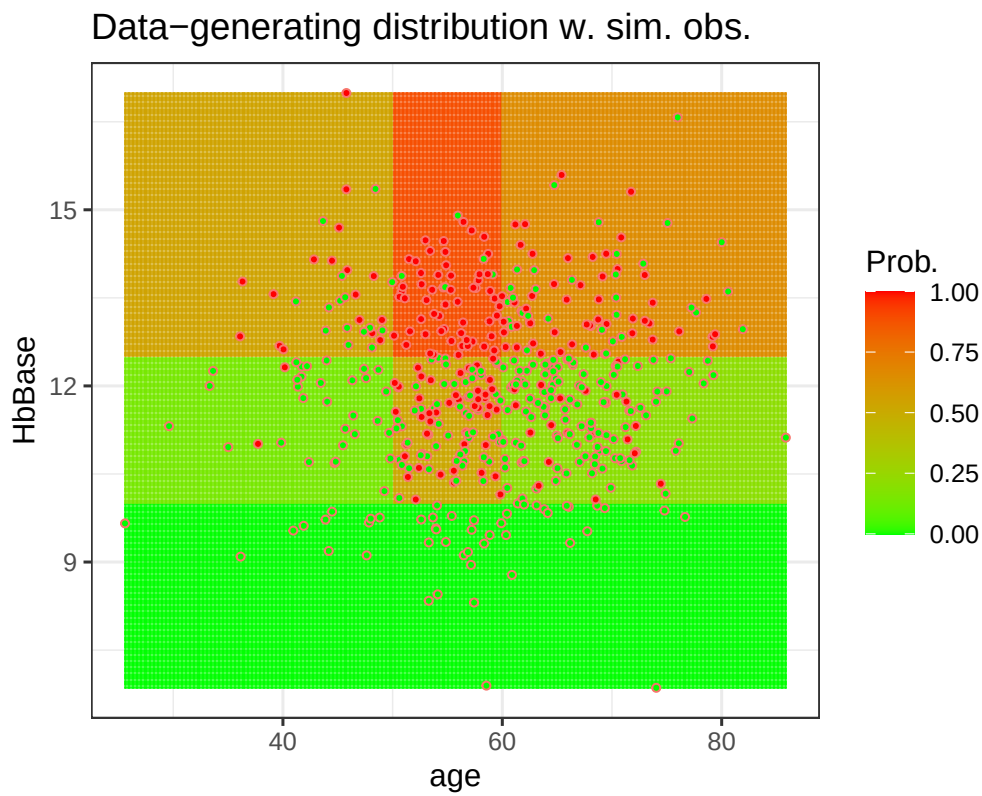
Sensitivity is a kind of variability

This is kind of a talk about variability in data.

Some variability is completely natural, while some variability is just noise/sampling variability.

We want to remove the possibility that the model fits the epistemic variability.





Random Forests

We will explore techniques used inside the forest.

1. Building multiple trees:
   - ~~Using resamples of data (Bootstrapping)~~
   - Using a subset of the covariates (Random subspace)

2. Averaging over the trees (Ensemble methods)

Different trees - better prediction

Random subspace is a method for making the trees even more different.

Random subspace selects only some of the covariates to split on per node. The method works in the same way as bootstrapping, but on the covariates instead of the observations.

Epo example

We saw in the previous example that HbBase will be the first splitting variable in most cases. This causes the trees to be correlated.

To remove some of the correlation we can bootstrap on the predictors.

```
colnames(Epo)[-1][1:4]
colnames(Epo)[-1][5:6]
```

```
[1] "age"    "sex"    "HbBase" "Treat"
[1] "Resection" "Receptor"
```

```
sample(colnames(Epo)[-1], 2, replace = TRUE)
```

```
[1] "Receptor" "Treat"
```

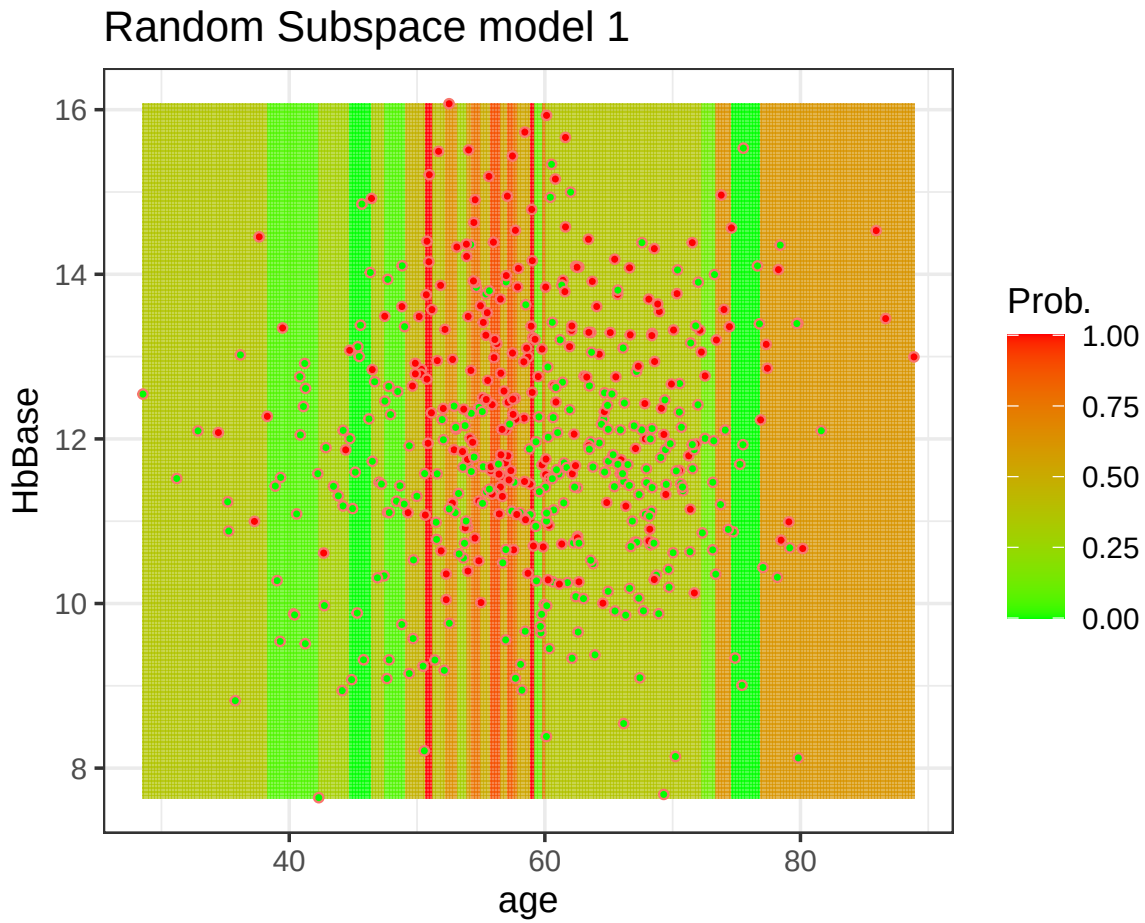
Epo simulation

Just as with bootstrapping, we can investigate this technique visually.

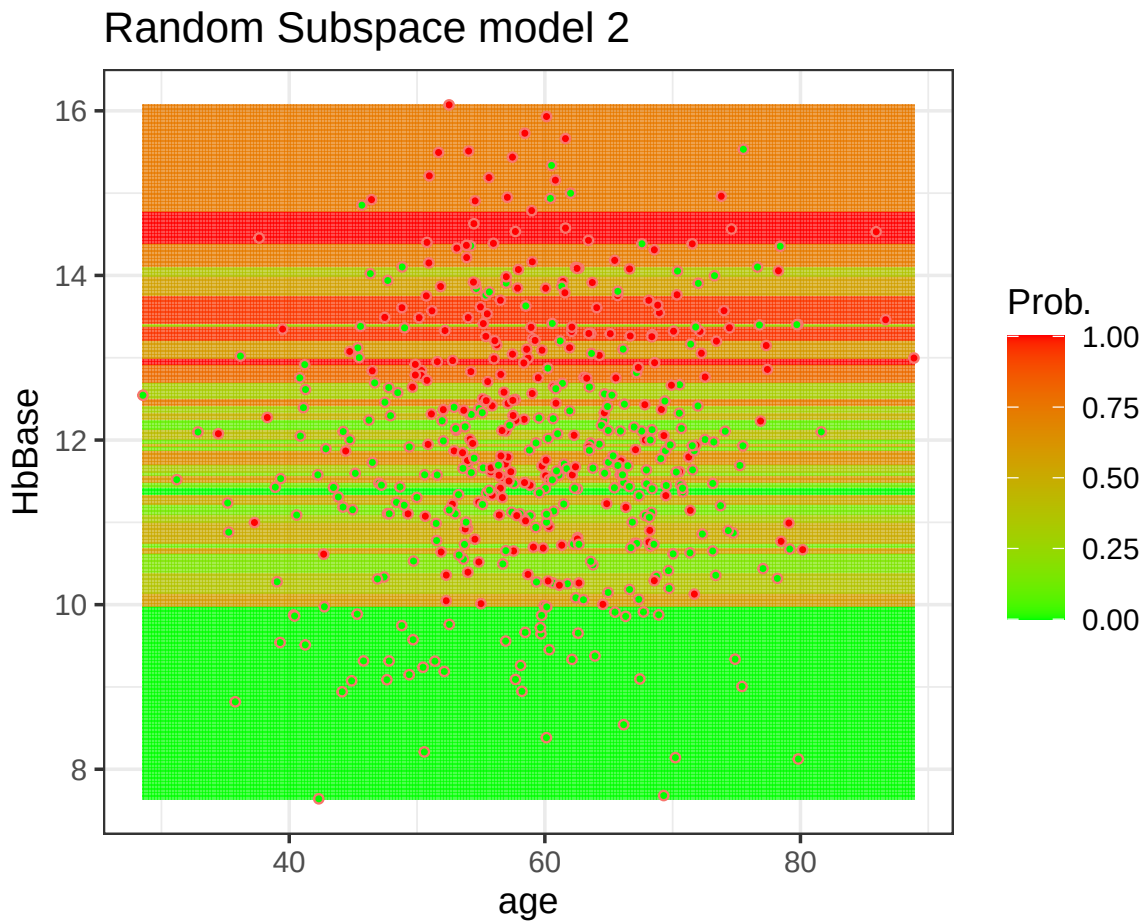
We keep working on the epo simulation, where we only had two predictors `HbBase` and `age`.

Random subspace 1 (greedy tree)

Which variable do we split on here?



Random subspace 2 (greedy tree)



Exercise 2 (part 1)

In this exercise, we will use the `randomForestSRC::rfsrc()` function to grow classification trees and random forests in the `Epo` data.

Preparation

```
library("randomForestSRC") # install.packages("randomForestSRC")
```

1. Bootstrap sample with replacement

Use the `sample` function to obtain a bootstrap sample from the `Epo` data. The number of observations in the bootstrap sample should be the same as the number of observations in the `Epo` data. How many unique patients (percentage) are in the bootstrap data set?

```
set.seed(5)
n <- nrow(Epo)
bootstrap.sample <- sample(1:n, n, replace = TRUE)
# number of unique patients
length(unique(bootstrap.sample))
# percentage
length(unique(bootstrap.sample))/n
```

Exercise 2 (part 2)

2. Three classifaction trees

Fit three classifaction trees (`ntree=1`) in the `Epo` data using the same seed:

- The first tree only uses the `HbBase` predictor
- The second tree only uses the age predictor
- The third tree uses both `HbBase` and age

```
tree1 <- rfsrc(Y ~ HbBase, data = Epo, ntree = 1, seed = 1)
tree2 <- rfsrc(Y ~ age, data = Epo, ntree = 1, seed = 1)
tree3 <- rfsrc(Y ~ age+HbBase, data = Epo, ntree = 1, seed = 1)
```

For each of the trees predict the chance of successful treatment for `newpatient`:

```
predict(tree1, newdata = newpatient, type = "response")$predicted
predict(tree2, newdata = newpatient, type = "response")$predicted
predict(tree3, newdata = newpatient, type = "response")$predicted
```

Are the predictions similar across the three models? Change the seed, re-fit the trees and predict `newpatient` again. Discuss the reasons why the predictions for the new patient of the 6 classification trees differ.

Exercise 2 (part 3)

3. Three random forests

- Now fit three random forest models each with 500 trees (`ntree = 500`) using all 6 predictors (`age`, `HbBase`, `Treat`, `Resection`, `Receptor`, `sex`). Fit two forests with the same seed and the last forest with a different seed.
- Predict the chance of successful treatment for `newpatient` using all three random forest models.
- Are the predictions more stable across the models compared to the classification tree models?
- Does it the seed influence the predictions of the random forest models with a clinically significant magnitude?

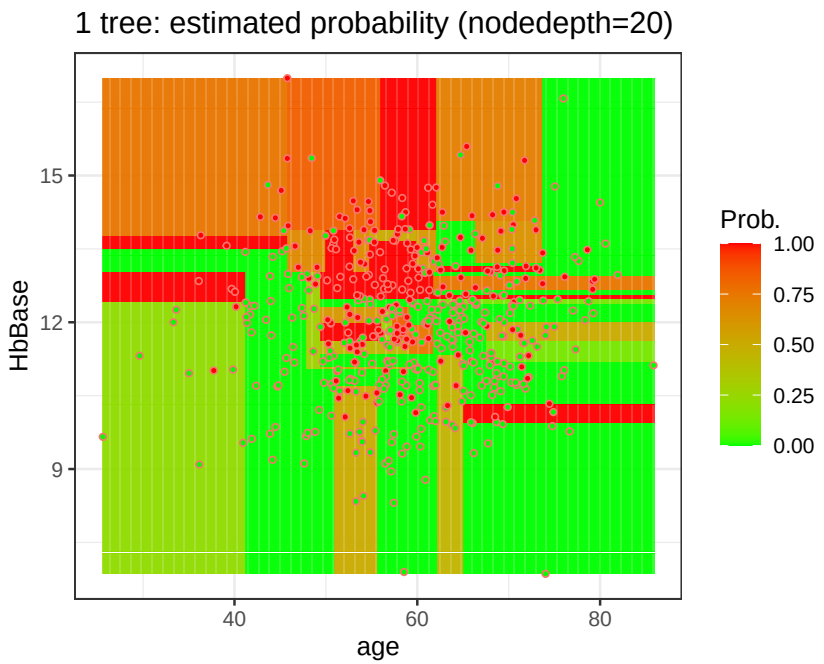
How to build a Random Forest

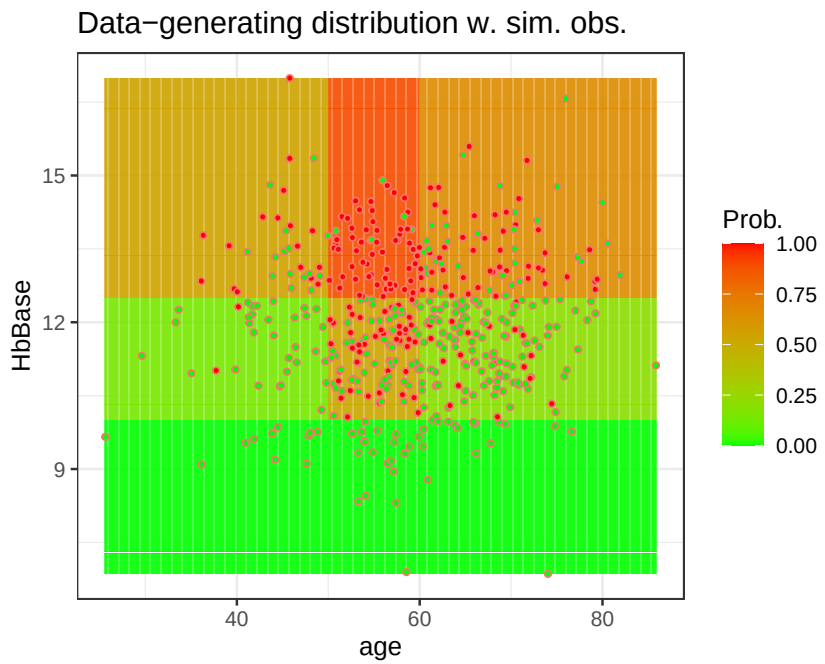
We will explore techniques used inside the forest.

1. Building multiple trees:
   - ~~Using resamples of data (Bootstrapping)~~
   - ~~Using a subset of the covariates (Random subspace)~~

2. Averaging over the trees (Ensemble methods)

Why not just a very deep tree?

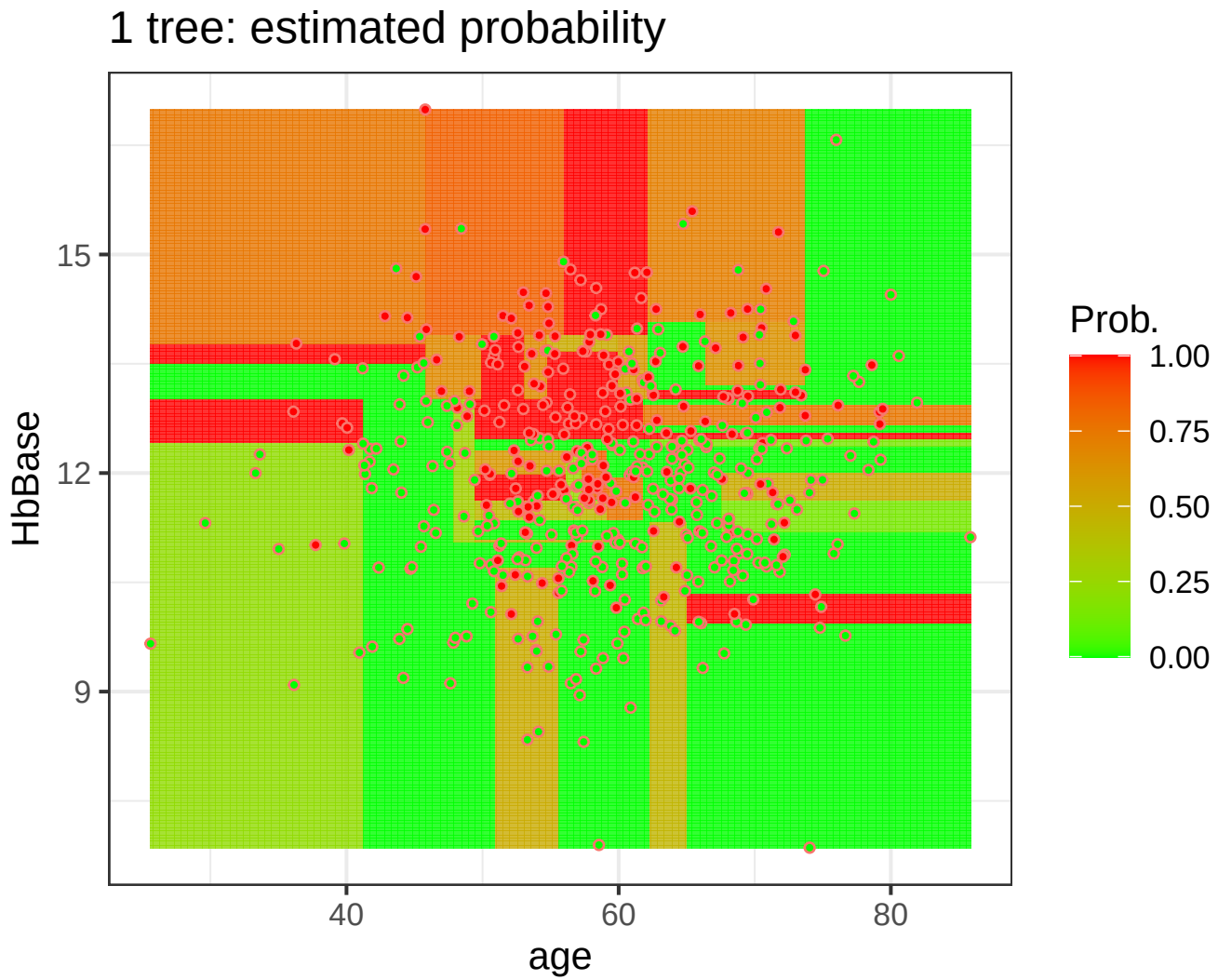




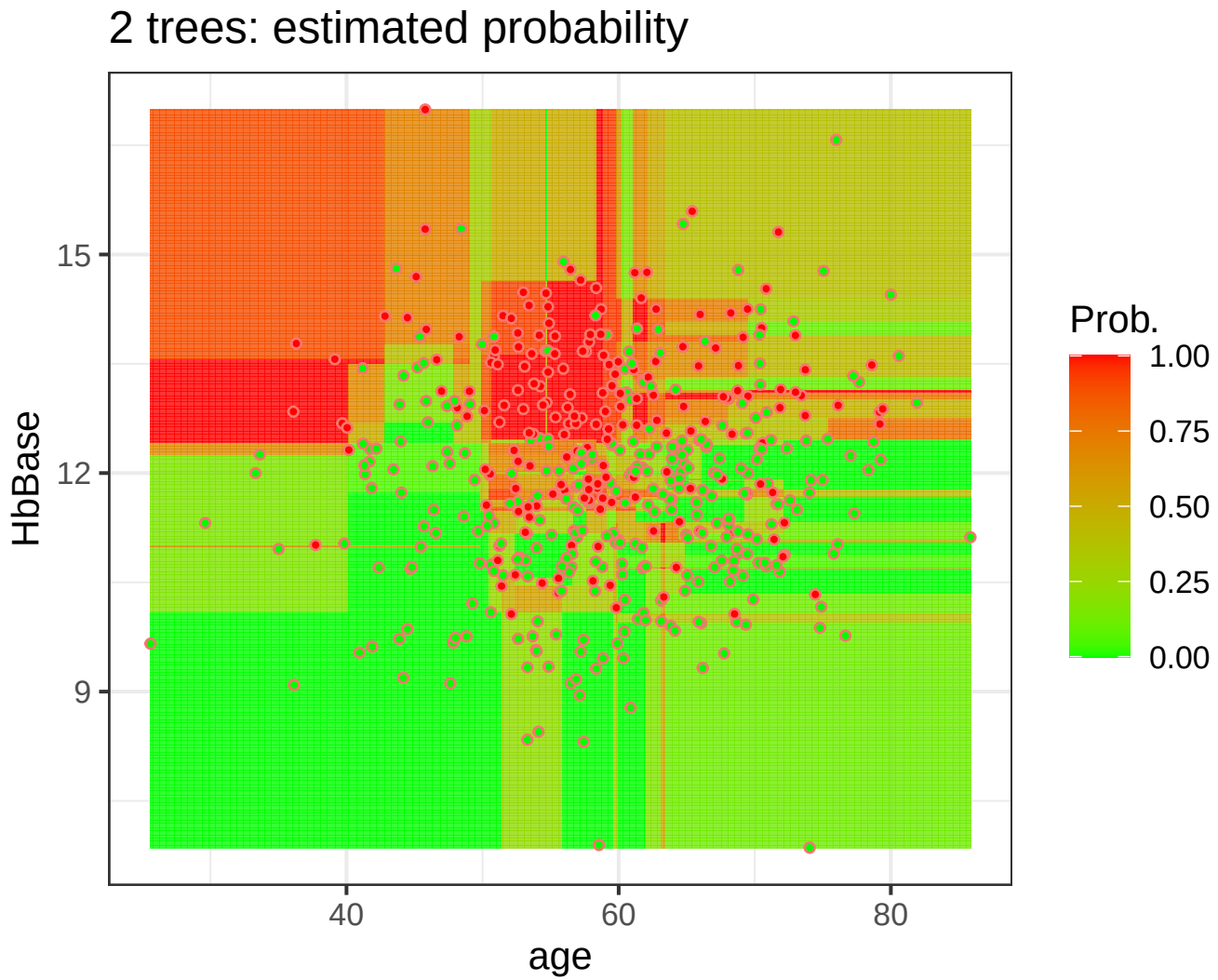
Averaging

Combine trees to fit $P(Y = 1 \mid \text{age}, \text{HbBase})$

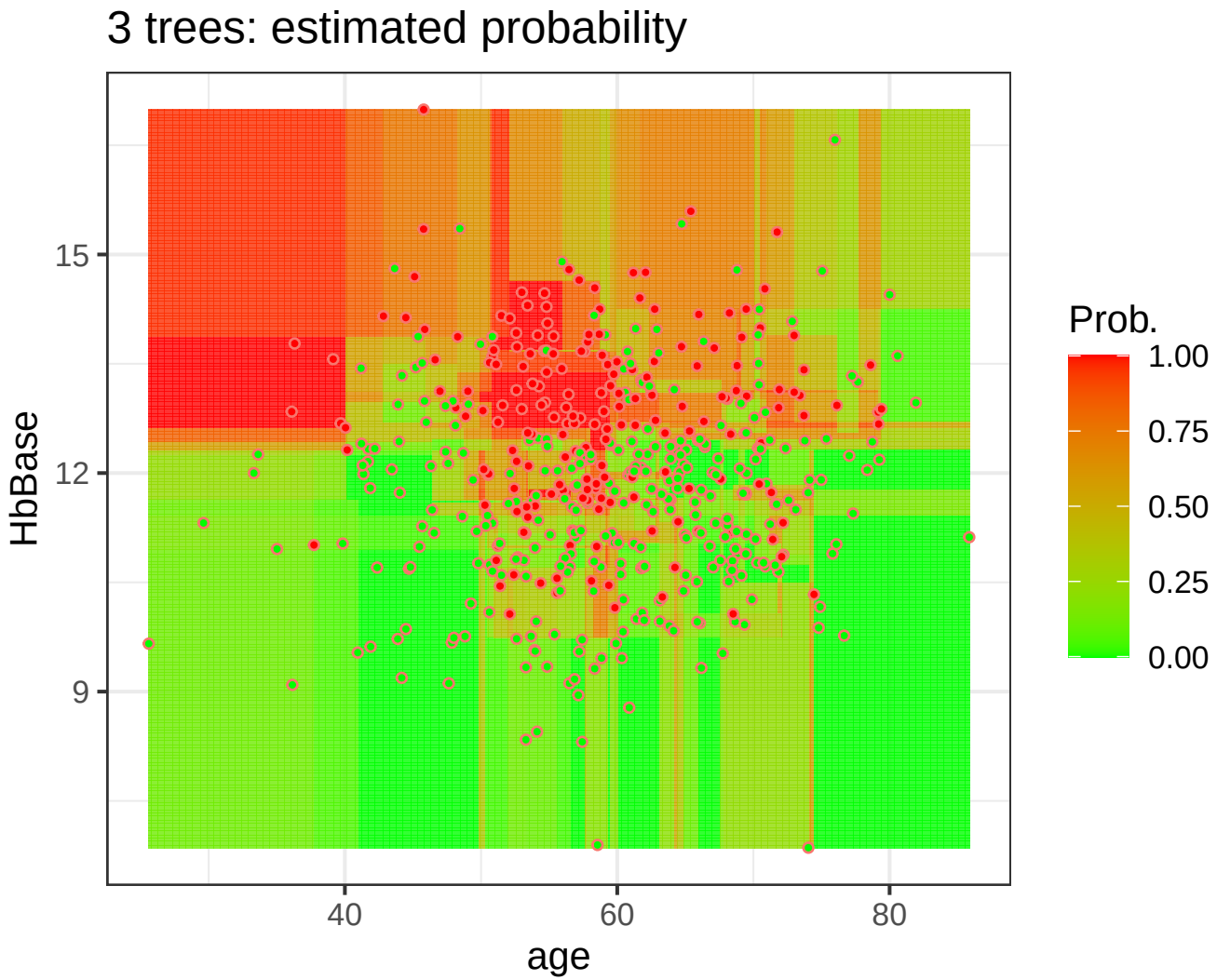
Averaging


1 tree: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

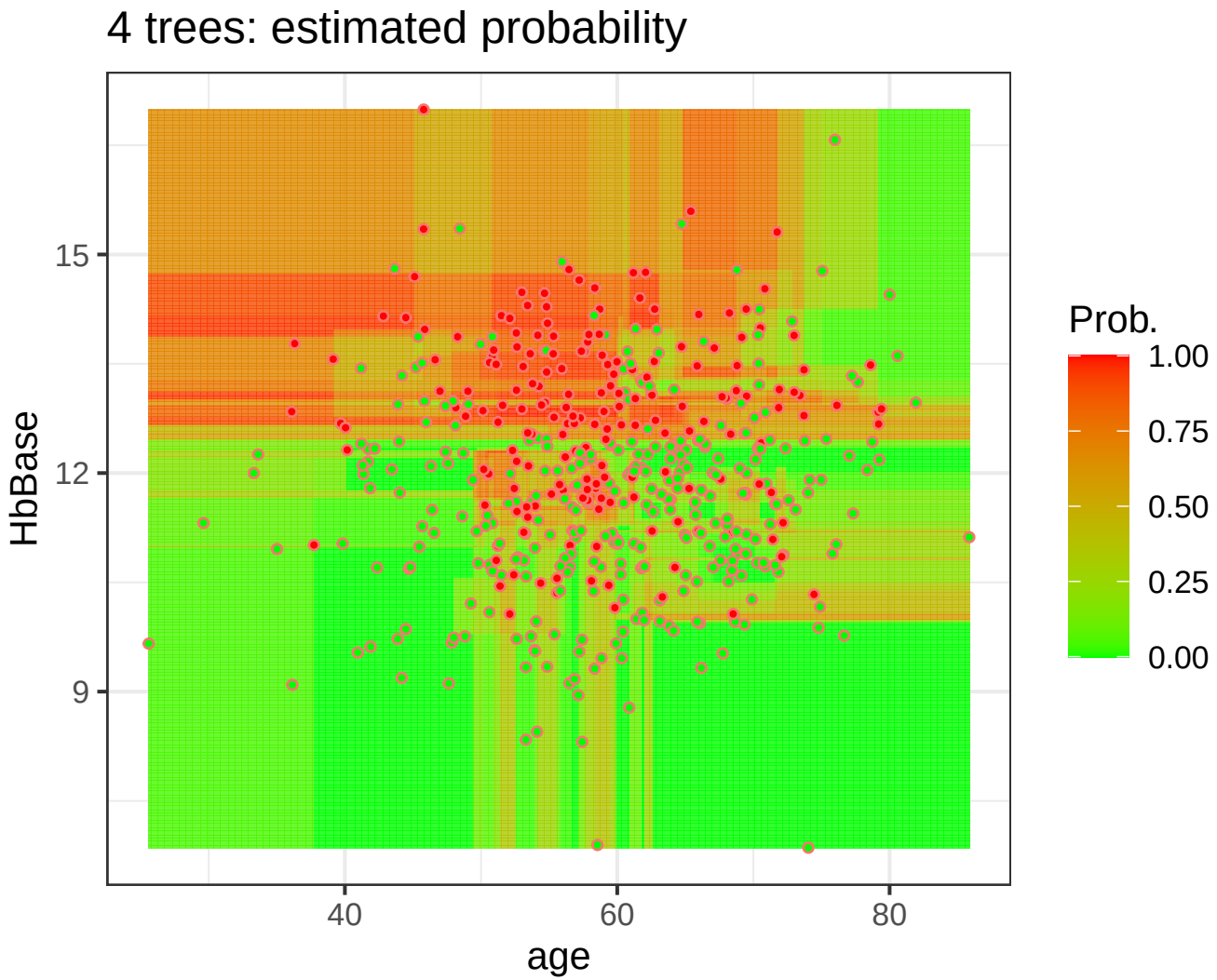
Averaging


2 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

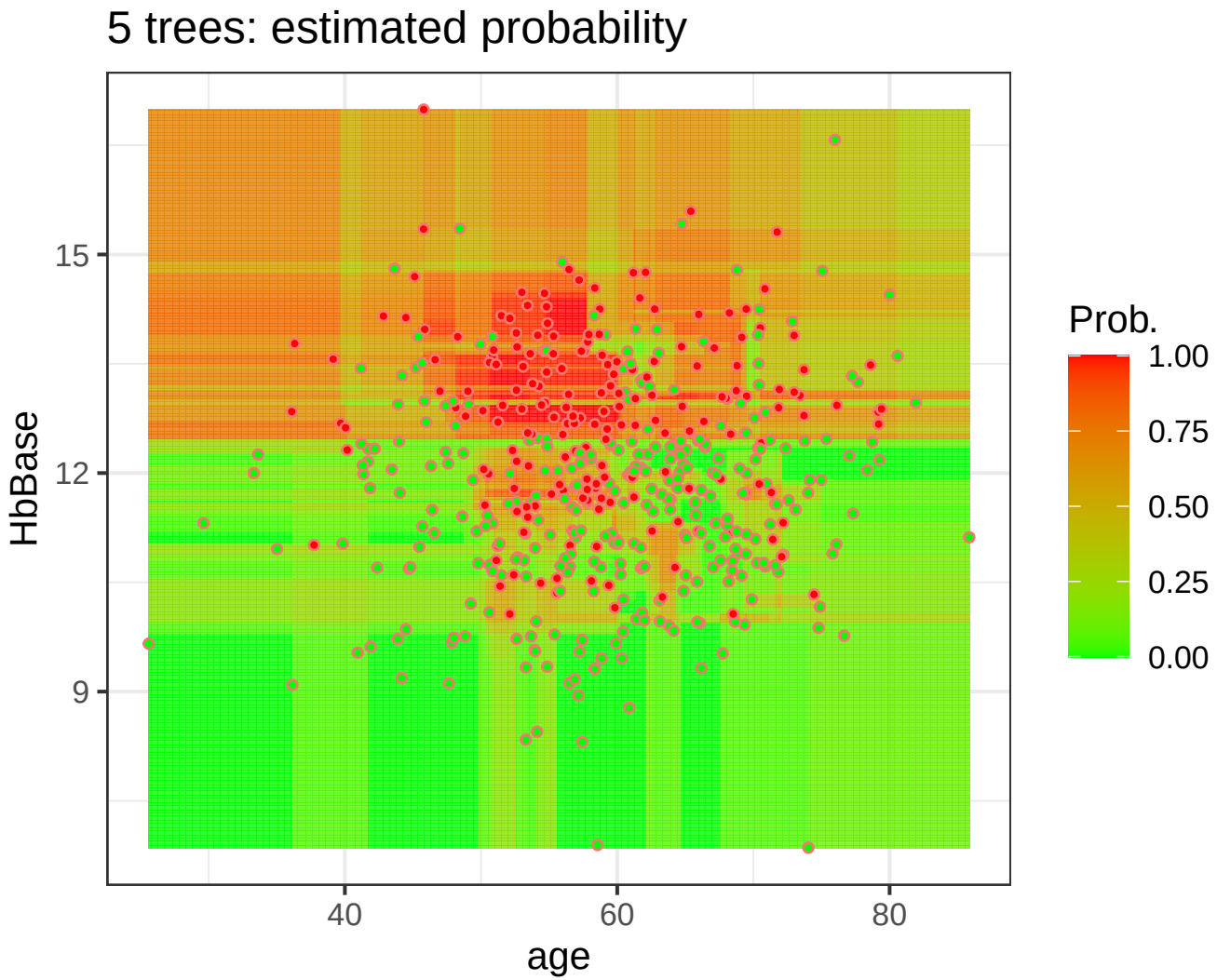
Averaging


3 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

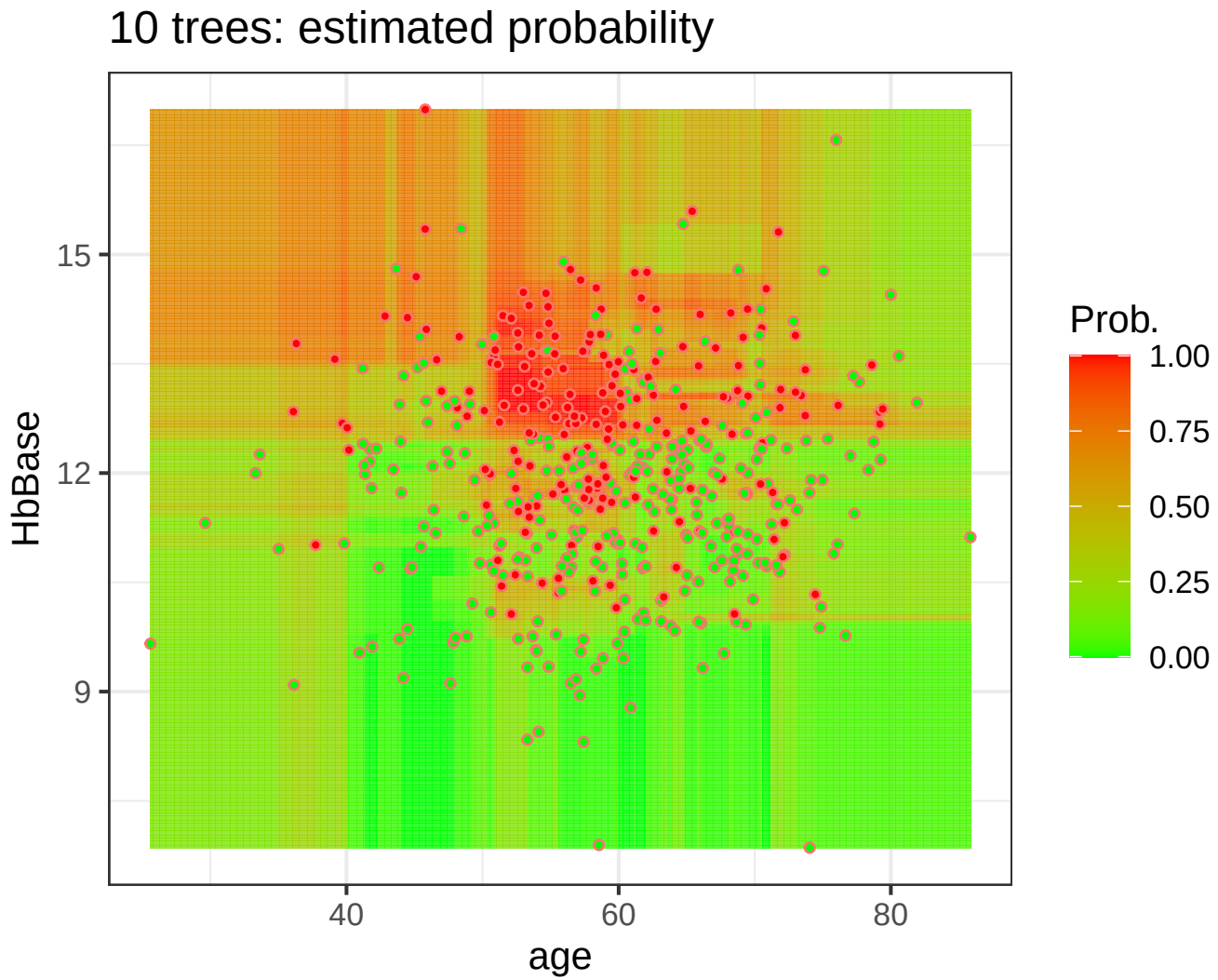
Averaging


4 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

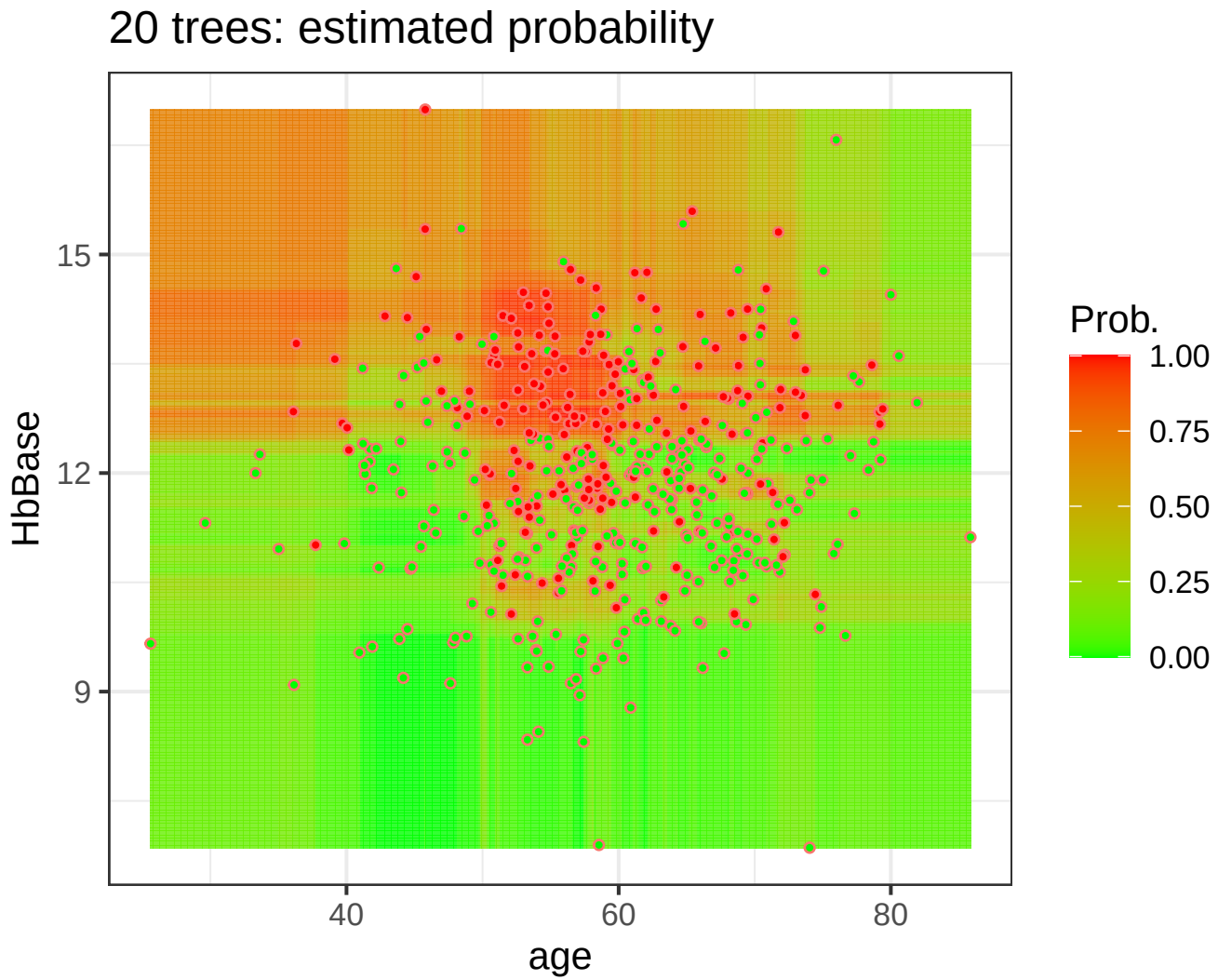
Averaging


5 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

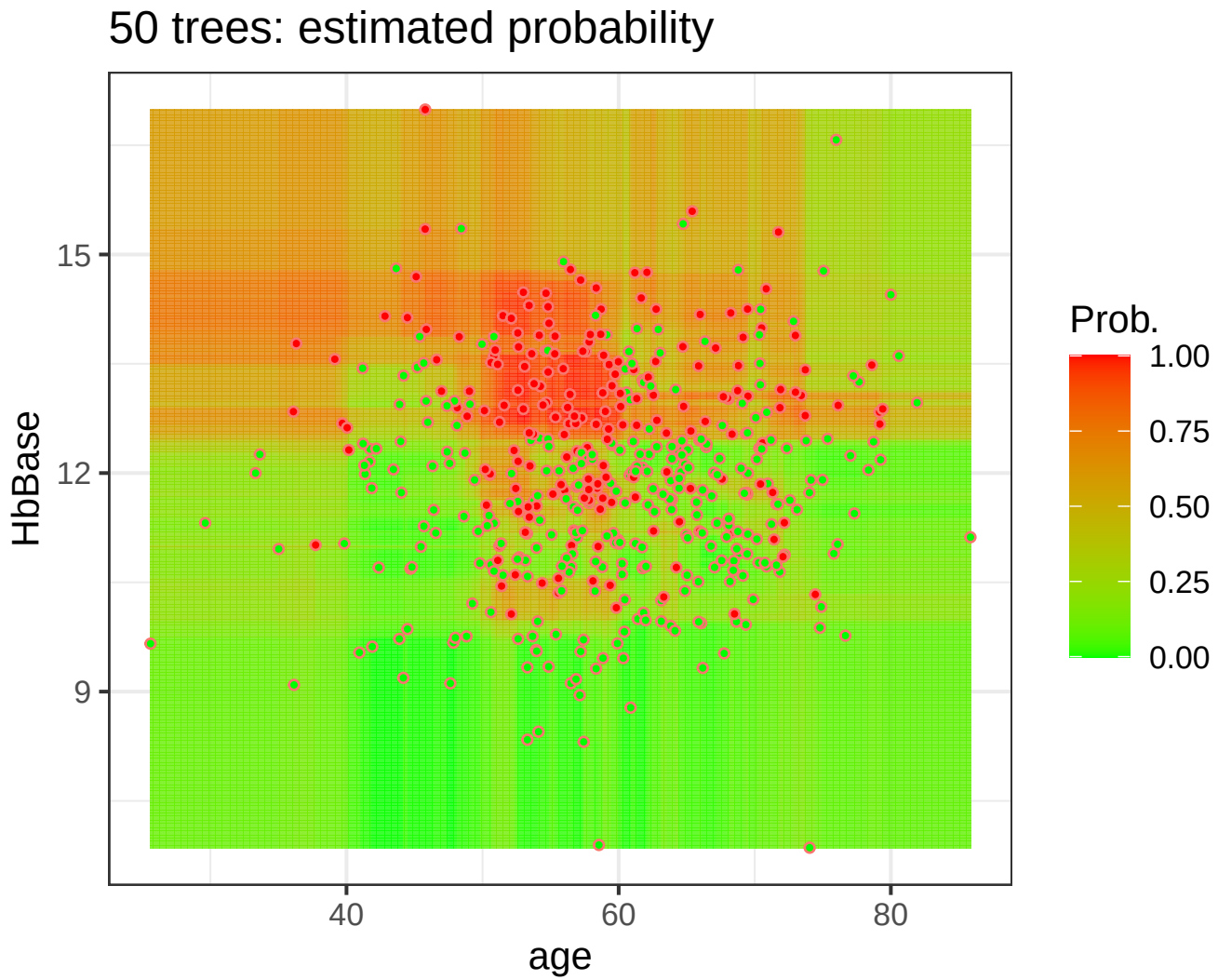
Averaging


10 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

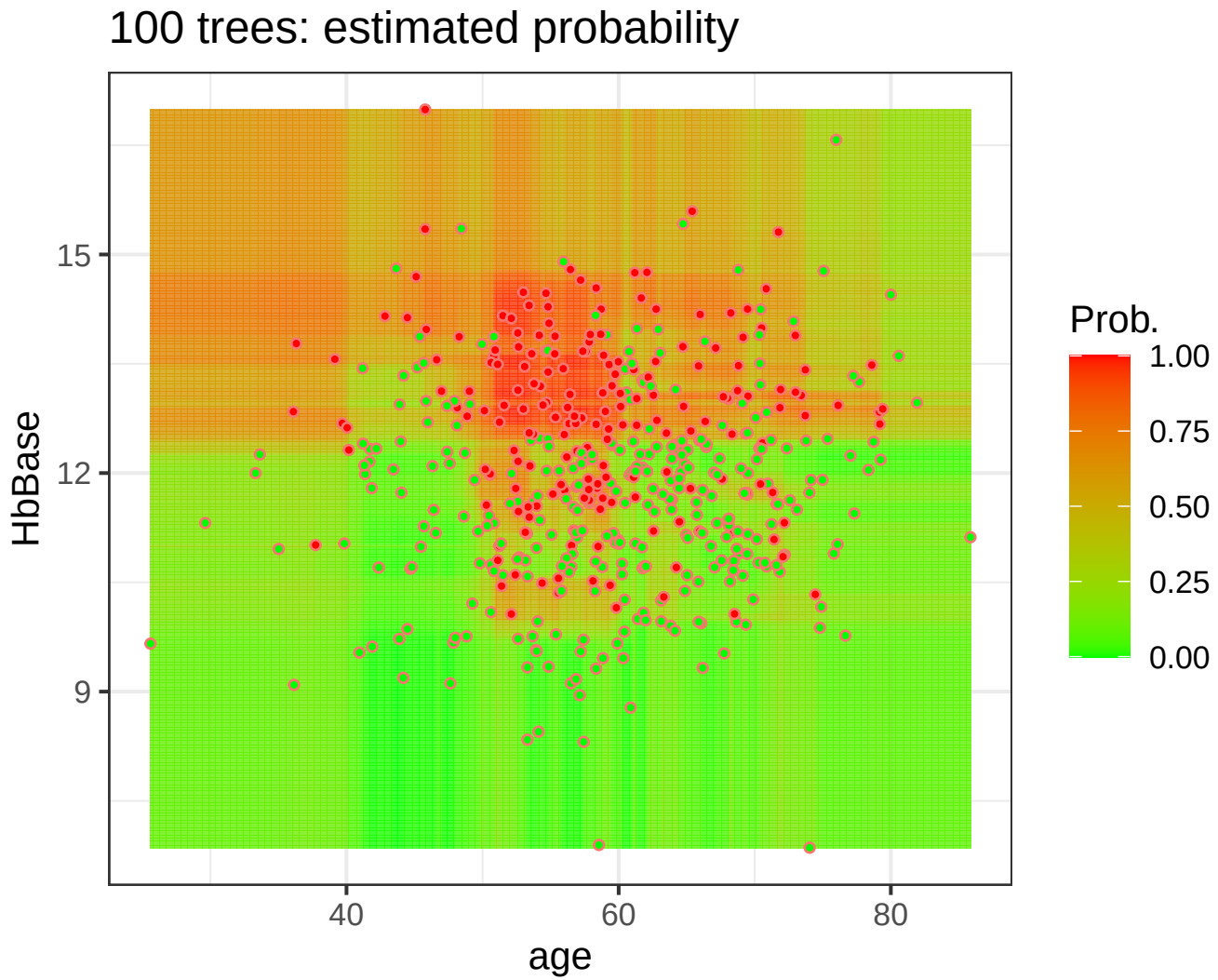
Averaging


20 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

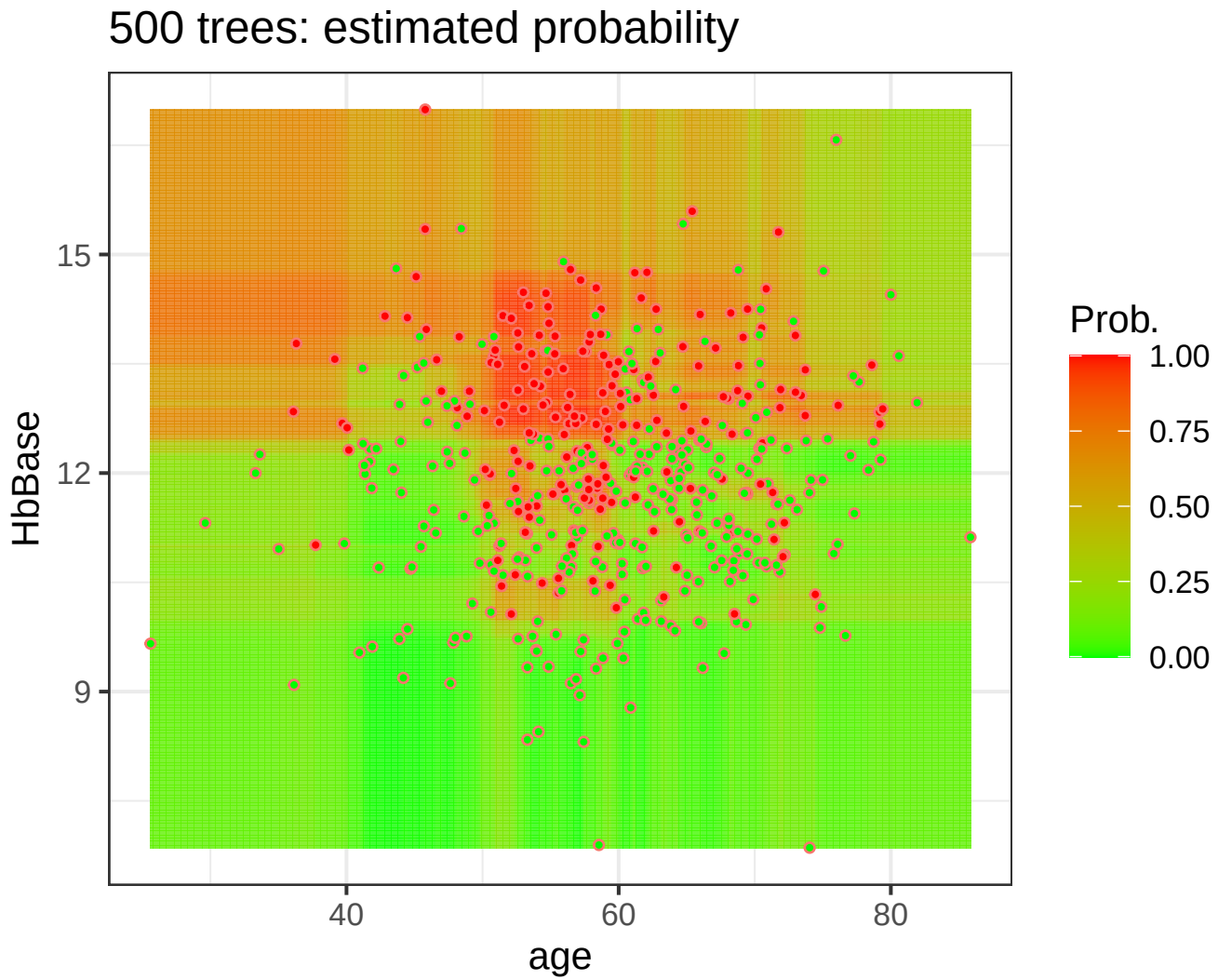
Averaging


50 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

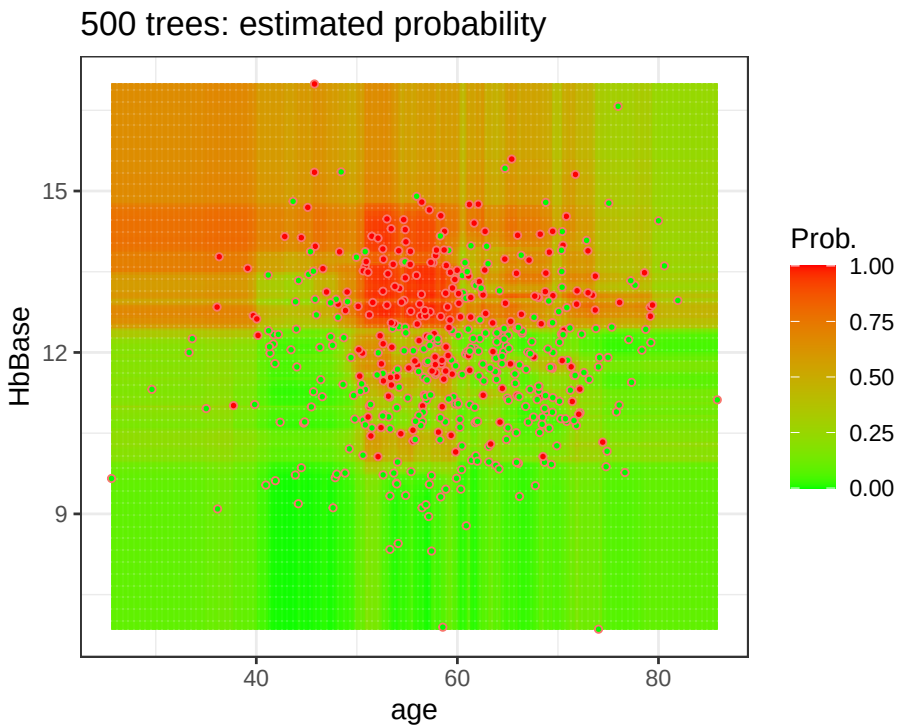
Averaging

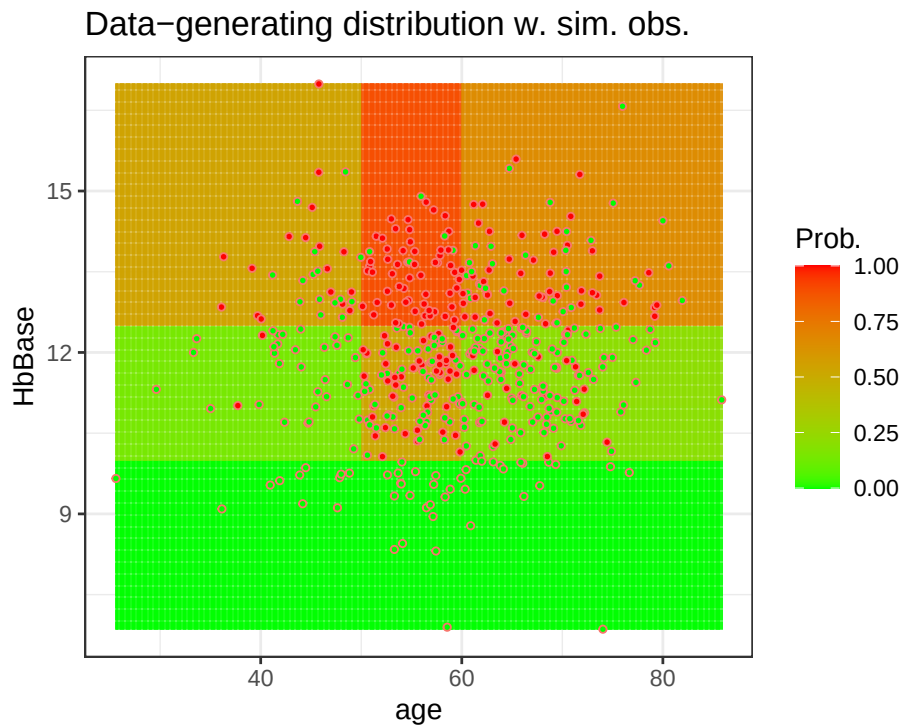

100 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

Averaging


500 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

Averaging


500 trees: estimated probability
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00


Data−generating distribution w. sim. obs.
HbBase
15
12
9
40
60
80
age
Prob.
1.00
0.75
0.50
0.25
0.00

Outline

- ~~Modeling cultures~~
- ~~Model selection~~
- ~~Decision trees~~
- ~~From trees to forests~~
  - ~~Random Forest and Machine Learning~~
  - ~~Bootstrapping~~
  - ~~Random subspace~~
  - ~~Ensemble methods~~
- Tuning random forests
- Variable importance

Fitting a random forest

We can now fit a forest with more confidence!

```
RFmodel <-
    rfsrc(Y~age+sex+HbBase+Treat+Resection+Receptor,
      data = Epo, mtry = 2, ntree = 500,
      nodesize = 5)
```

And get predictions:

```
round(RFmodel$predicted[1:5],3)
```

```
[1] 0.075 0.066 0.956 0.916 0.052
```

Prediction 1 vs prediction 2

But another person also fit a random forest to the Epo data, but with other hyperparamters. . .

```
RFmodel2 <-
    rfsrc(Y~age+sex+HbBase+Treat+Resection+Receptor,
      data = Epo, mtry = 3, ntree = 100,
      nodesize = 2)
```

And got predictions:

```
round(RFmodel2$predicted[1:5],3)
```

```
[1] 0.013 0.007 0.987 0.994 0.000
```

Evaluation

Evaluating Random Forest models = evaluating the predictions

What is the purpose of the predictions?

What exactly is available to us to make evaluations?

Prediction error and predictive accuracy

We know the history of our data

We observe for each patient i in the data:

$$Y_i = \begin{cases} 1 & \text{treatment successful} \\ 0 & \text{treatment failed} \end{cases}$$

Based on the history, we fit a model (e.g. using Random Forest)

We hence also have an estimate for the patient i:

$$\widehat{P}_i = P(Y_i = 1 \mid \text{age}_i, \text{HbBase}_i)$$

based on the $i'th$ patient's observed age (age_i) and Baseline hemoglobin value (HbBase_i)

Predictive accuracy

We can evaluate the predictions in context to the observed outcome

Patient no.	Y Treatment successful	Predicted probability
1	0	$\widehat{P}_1$
2	0	$\widehat{P}_2$
3	1	$\widehat{P}_3$
4	1	$\widehat{P}_4$
5	0	$\widehat{P}_5$
6	1	$\widehat{P}_6$
7	1	$\widehat{P}_7$
.	.	.
.	.	.

Prediction error and predictive accuracy

Prediction error is measured in terms of some distance[8] between:

1) the observed outcome: Y_i

2) and the predicted probability: $\widehat{P}_i = \widehat{P}(Y_i = 1 \mid \text{age}_i, \text{HbBase}_i, \ldots)$

One example of a loss function is the squared error loss:

$$L(Y_i, \widehat{P}_i) = (Y_i - \widehat{P}_i)^2$$

[8] Measured in terms of a *loss function*

Brier score

Evaluating the model can then be done using the Brier score.

Let n be the number of observations,

$$\text{Brier score} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{P}_i)^2$$

As we are dealing with binary data Y_i and predicted probabilities $\widehat{P}_i$, the Brier score is between 0 and 1.

The smaller the Brier score, the better the prediction.

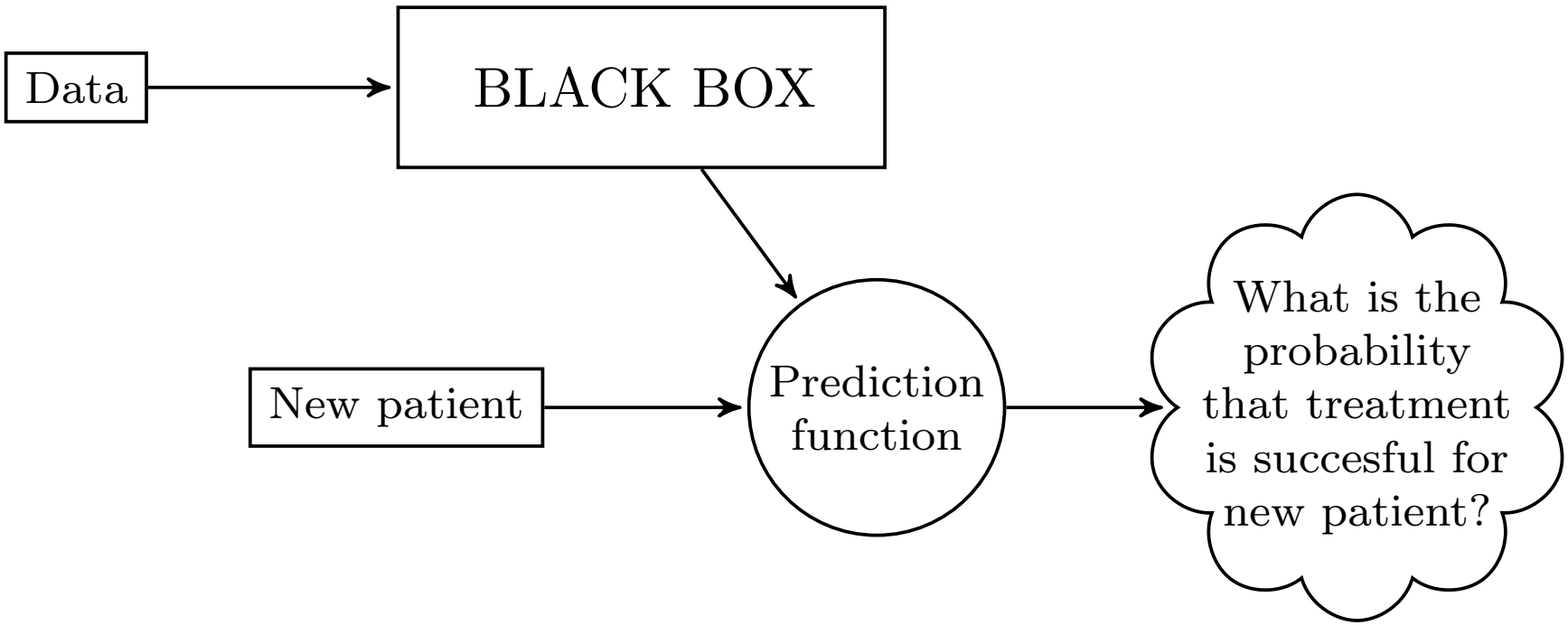
Other criteria

There are other scores to look at than the Brier score.

Other popular choices include AUC (Area under the curve). This method works well for discrimination between patients - hence it is evaluating the ranking of the patients.

This kind of loss function is not a calibrating loss function - it does not target getting the correct probabilities.

Predictive for who?



Easy to be predictive when knowing the truth

We can evaluate the predictions in context to the observed outcome

Patient no.	Y Treatment successful	Predicted probability
1	0	P_1
2	0	P_2
3	1	P_3
4	1	P_4
5	0	P_5
6	1	P_6
7	1	P_7
.	.	.
.	.	.

But that is a little like cheating

Not easy when we do not know the truth

We will not know the outcome for the patients in the future.

Patient no.	Y Treatment successful	Predicted probability
1001	?	P_{1001}
1002	?	P_{1002}
1003	?	P_{1003}
1004	?	P_{1004}
1005	?	P_{1005}
1006	?	P_{1006}
1007	?	P_{1007}
.	.	.
.	.	.

We want our model to perform well on people we have not observed

Coercing data to handle the future

How do we coerce our data to resemble this?

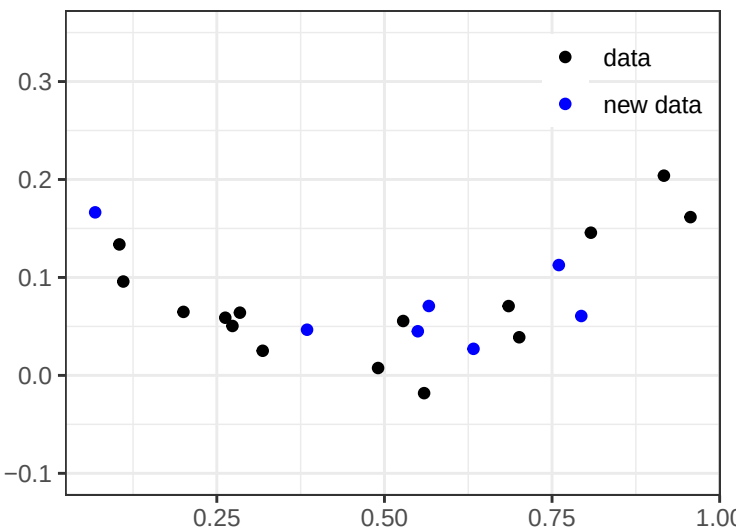
Machine learning 101

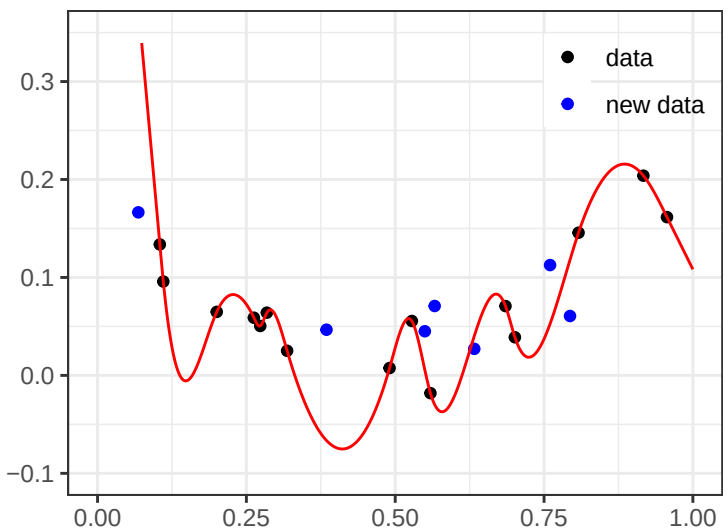
To measure the prediction error correctly, we cannot train the model and assess the model on the same data

Why? Overfitting!

Overfitting happens when a model learns the detail and noise in the data too well so that it negatively impacts the performance of the model on new data.

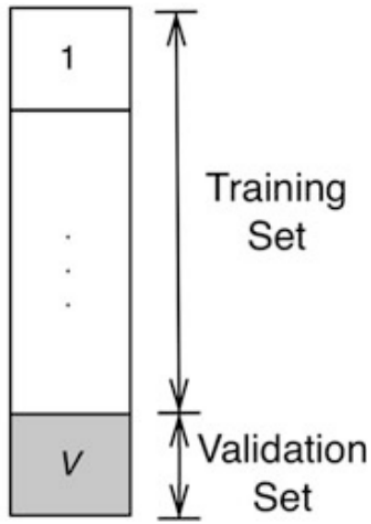
Evaluating a model on the same data results in overfitting.





Sample splitting

To evaluate predictions on something we do not train on, we can use sample splitting:



1. I create and fit my model on the training data: P^{train}
2. I check the quality of my model on the validation data
   - Average of $L(Y_i, \hat{P}_i^{\text{train}})$ in validation sample

Epo: Sample splitting

Now on real data! (Not simulation).
Let us create some training data:

Fix seed:

```
set.seed(5)
```

Take 10 % of original data to be our validation set:

```
val.set <- sample(x = 1:n, size = n/10, replace=FALSE)
```

The rest comprise our training data:

```
train.set <- (1:n)[!(1:n) %in% val.set]
```

Epo: Training

Fit 1 tree on the training data:

```
tree1.train <-
    rfsrc(Y~age+sex+HbBase+Treat+Resection+Receptor,
      data = Epo[train.set,],
      ntree=1, seed=1)
```

Or . . . fit a forest of 100 trees on the training data:

```
forest.train <-
    rfsrc(Y~age+sex+HbBase+Treat+Resection+Receptor,
      data = Epo[train.set,],
      ntree=100, seed=1)
```

Epo: Predicting

Predict from the tree model on the validation set:

```
tree1.val <- predict(model = tree1.train,
                     newdata = Epo[val.set,],
                     type = "response")$predicted
```

Predict from the forest model on the validation set:

```
forest.val <- predict(model = forest.train,
                      newdata = Epo[val.set,],
                      type = "response")$predicted
```

Epo: Evaluating

We define the loss function (Brier score):

```
loss.fun <- function(Y, Phat) mean((Y - Phat)^2)
```

Now we can compare performance:

```
print(rbind(
    "1 tree " = loss.fun(Y = Epo[val.set, ]$Y,
              Phat = tree1.val),
    "forest " = loss.fun(Y = Epo[val.set, ]$Y,
              Phat = forest.val)))
```

```
              [,1]
1 tree   0.1443149
forest   0.0726101
```

Which one seems to perform best?

Exercise 3 (part 1)

In this exercise, we will investigate the Brier score using the Epo data set. We want to evaluate which of the models performs the best. We also get more knowledge about how to use the Brier score.

1. 50% model

In the Epo data set sample non-overlapping training and validation data sets. The validation set should include 1/10 of the total sample size.

```
set.seed(5) # set seed
n <- dim(Epo)[1]
val.set <- sample(x = 1:n, size = n/10, replace=FALSE) # create val.
    indicator
train.set <- (1:n)[!(1:n) %in% val.set] # create train indicator
Epo.train <- Epo[train.set,]
Epo.val <- Epo[val.set,]
```

We start with a prediction model which predicts 50% chance for all patients. What is the Brier score in the validation data set using this model?

```
Epo.val$prediction <- 0.5 # set prediction to 50%
loss.fun <- function(Y, Phat) mean((Y - Phat)^2) # define the Brier Score
loss.fun(Epo.val$Y, Epo.val$prediction)
```

Would we ever use a prediction model with a larger Brier score than this model?

Exercise 3 (part 2)

2. Random forest model

On the training data, fit a random forest using all predictors (remember to set a seed). Once you have fitted your random forest, you can add the predictions to the validation data:

```
Epo.val$prediction <- predict(YourRandomForestModel,
                    newdata = Epo.val, type = "reponse")$predicted
```

Calculate the Brier score in the validation data. Based on the Brier score, is the random forest a better prediction model than the simple 50% model?

Exercise 3 (part 3)

3. Another random forest model

Use a new seed to create a new split into training (9/10) and validation data (1/10). Refit your random forest and re-calculate the Brier score. Can you compare this random forest model to the model fitted in the previous training data?

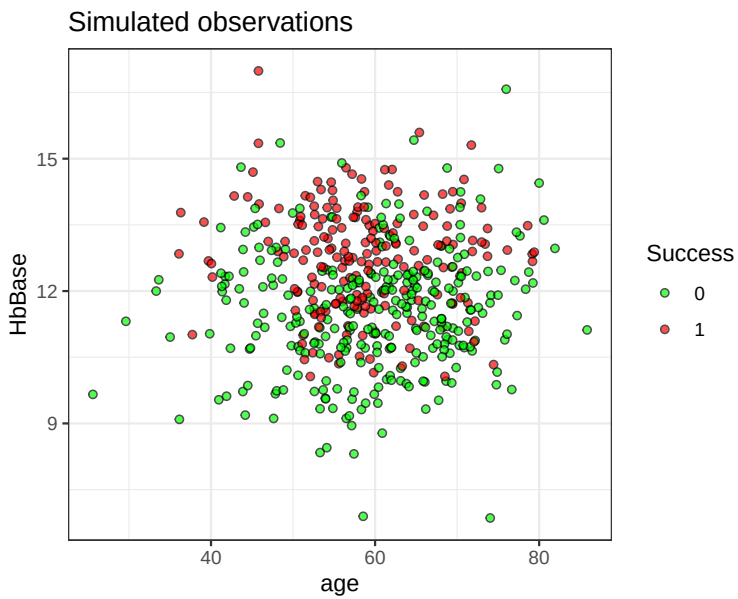
4. Smaller is better

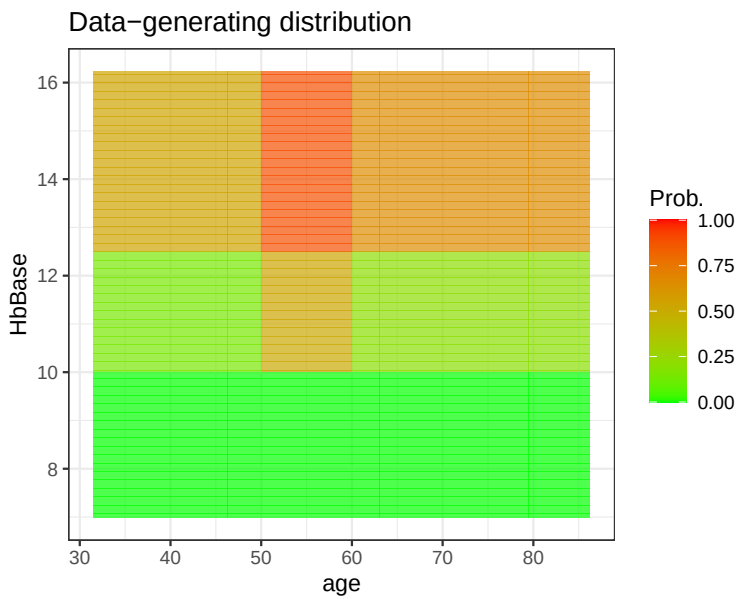
We know that the smaller the Brier score, the better the prediction model. Discuss, if it is possible to have a Brier score of 0. When is the Brier score exactly 0 (you can look at the formula)?

Exercise 3 (part 4)

5. Brier score of 0?

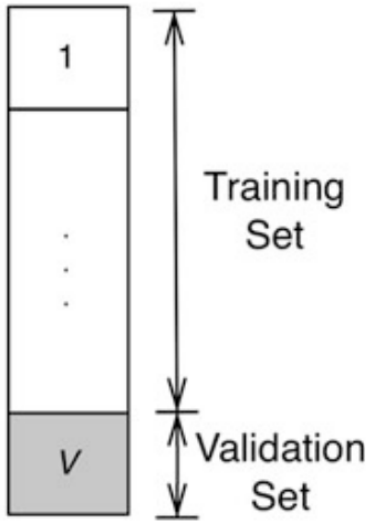
In the simulated data example, we know the data generating model. Explain the figure below: why could we never get a Brier score of exactly 0 for the data-generating model.





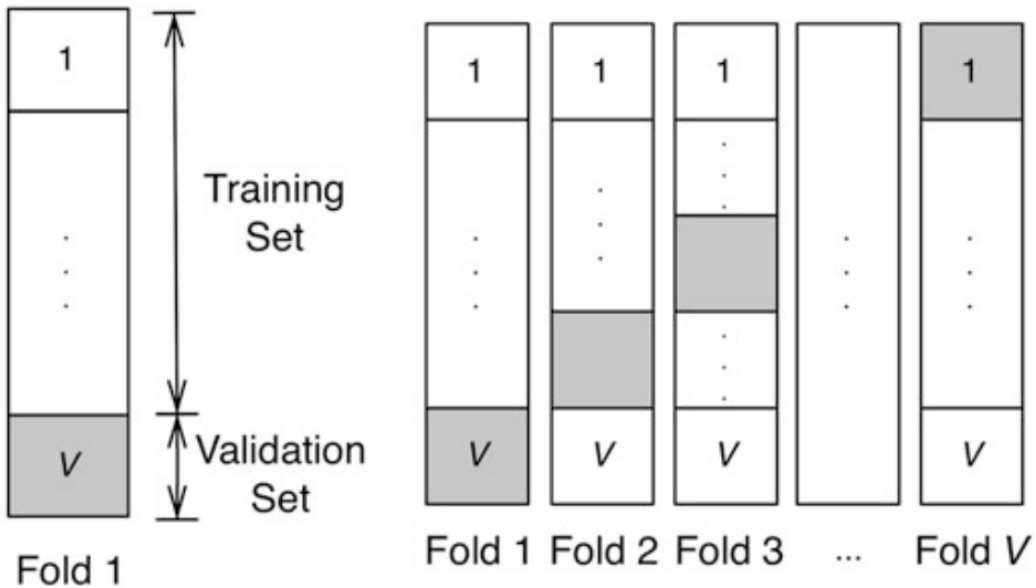
V fold cross-validation

In practice, the splitting of data is not done once



V fold cross-validation

In practice, the splitting of data is not done once



. . . but several times: This is called V-fold cross-validation

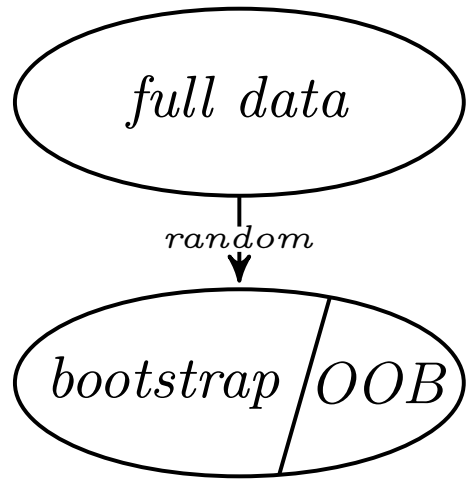
That almost sounds like the Bootstrapping we just learned about?

Bootstrapping

It is almost the same and it is done for almost the same reasons!

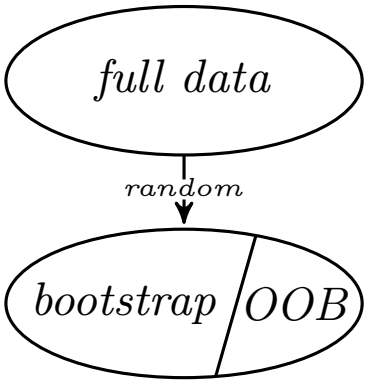
So when we have validation data and we use random forest, we try to reduce overfitting in two ways.

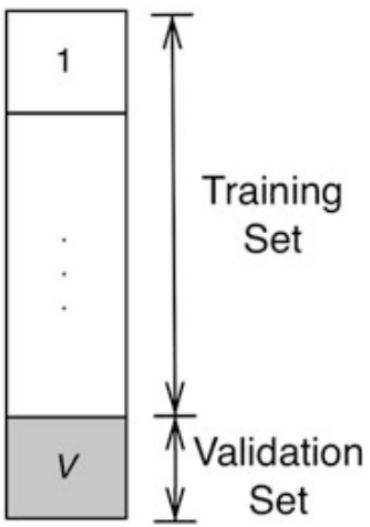
- We call those not fitted in the model given the specific bootstrap for OOB (Out-of-bag)
- The rest of the subjects are called in-bag



We can use the out of bag data (OOB) for tuning our model

- The oob prediction for patient i only uses the trees built on bootstrap samples where patient i was left out of bag.



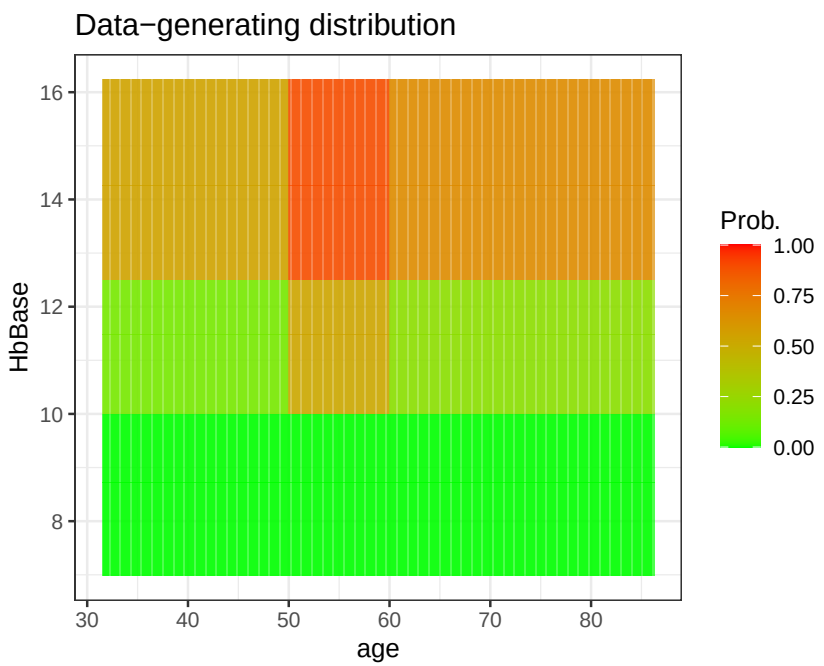


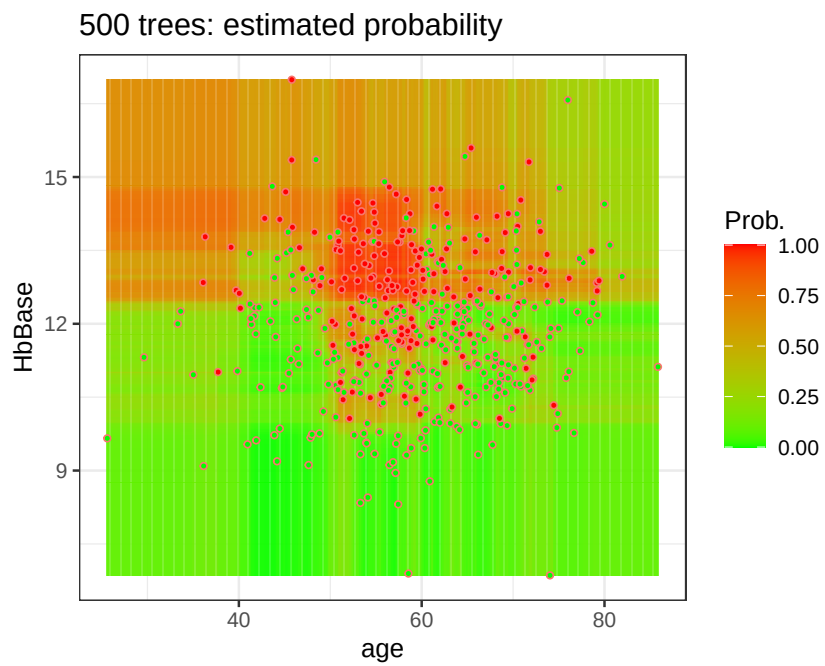
- The out of bag prediction error is estimated by:

$$\widehat{\text{error}}_{\text{oob}} = \frac{1}{n}\sum_{i=1}^{n} L(Y_i, \hat{P}_i^{\text{oob}})$$

Picking the random forest model

The random forest algorithm automatically detects nonlinear effects, complex interactions, . . .





Picking the random forest model

But the algorithm involves some choices: Hyperparameters

- These can be tuned and lead to different results
- These can be tuned to optimize predictive performance

Tuning intro

There are mainly three variables that will be used

1. The number of predictors to randomly select at each split (`mtry`)
2. The total number of trees in the ensemble (`ntree`)
3. The minimum number of leaf node size (`nodesize`)

```
RFmodel <-
    rfsrc(Y~age+sex+HbBase+Treat+Resection,
      data = Epo, mtry = 2, ntree = 500,
      nodesize = 5)
```

Trial-and-error

We are going to experiment with a lot of different sets of hyperparameters to find the best model fit.

Finding the best hyperparameters is a very experimental process, while there can be some logic around selecting your hyperparameters, it is standard to search for the best via e.g. grid-search.

How informative is the predictors that we have in our model? If the majority is considered strong predictors of outcome, then the number can be small.

Number of random features

Epo example

We saw in the previous example that HbBase will be the first splitting variable in most cases. This causes the trees to be correlated.
To remove some of the correlation we can bootstrap on the predictors.

```
colnames(Epo)[-1][1:3]
colnames(Epo)[-1][4:6]
```

```
[1] "age"    "sex"    "HbBase"
[1] "Treat"      "Resection" "Receptor"
```

```
sample(colnames(Epo)[-1], 2, replace = TRUE)
```

```
[1] "Receptor" "Treat"
```

Trying different mtry

Trying with mtry = $\{1, 2, 3, 4, 5, 6\}$.

```
rf_mt1 <-
  rfsrc(Y~age+sex+HbBase+Treat+Resection+Receptor,
    data = Epo, mtry = 1, seed = 1)
rf_mt2 <-
  rfsrc(Y~age+sex+HbBase+Treat+Resection+Receptor,
    data = Epo, mtry = 2, seed = 1)
rf_mt3 <-
  rfsrc(Y~age+sex+HbBase+Treat+Resection+Receptor,
    data = Epo, mtry = 3, seed = 1)
```

etc.

Trying different `mtry`

Looking at the out of bag Brier score we get:

```
         OOB Brier score
mty = 1        0.1370008
mty = 2        0.1047253
mty = 3        0.1029690
mty = 4        0.1034286
mty = 5        0.1058208
mty = 6        0.1059194
```

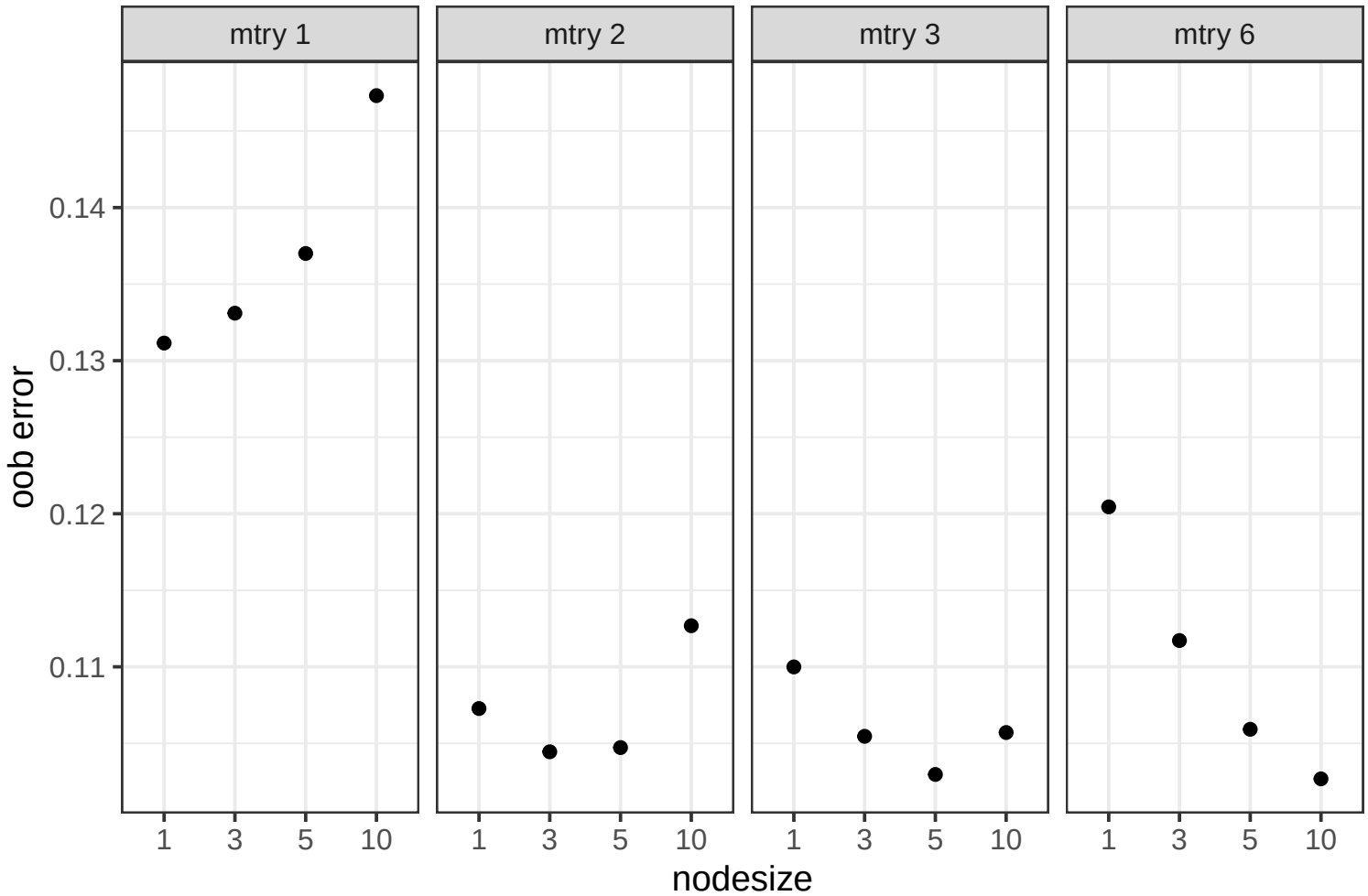
Combining with nodesize

nodesize affects how deep the individual decision trees can be. A small nodesize will return deeper trees.

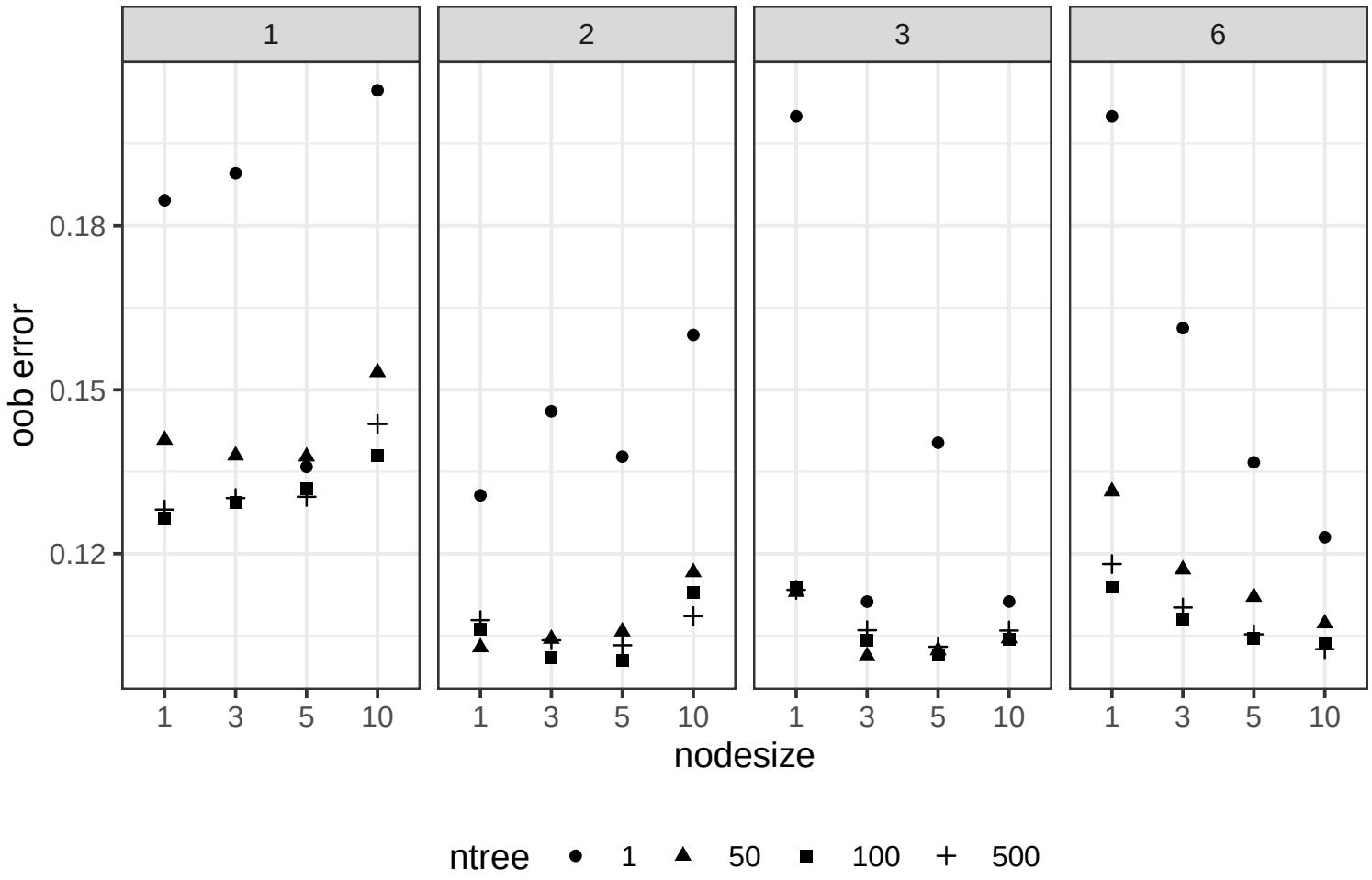
```
expand.grid(mtry = c(1,2,3,6),
            nodesize = c(1,3,5,10))[1:10,]
```

```
   mtry nodesize
1     1        1
2     2        1
3     3        1
4     6        1
5     1        3
6     2        3
7     3        3
8     6        3
9     1        5
10    2        5
```

Tuning `mtry` and `nodesize`



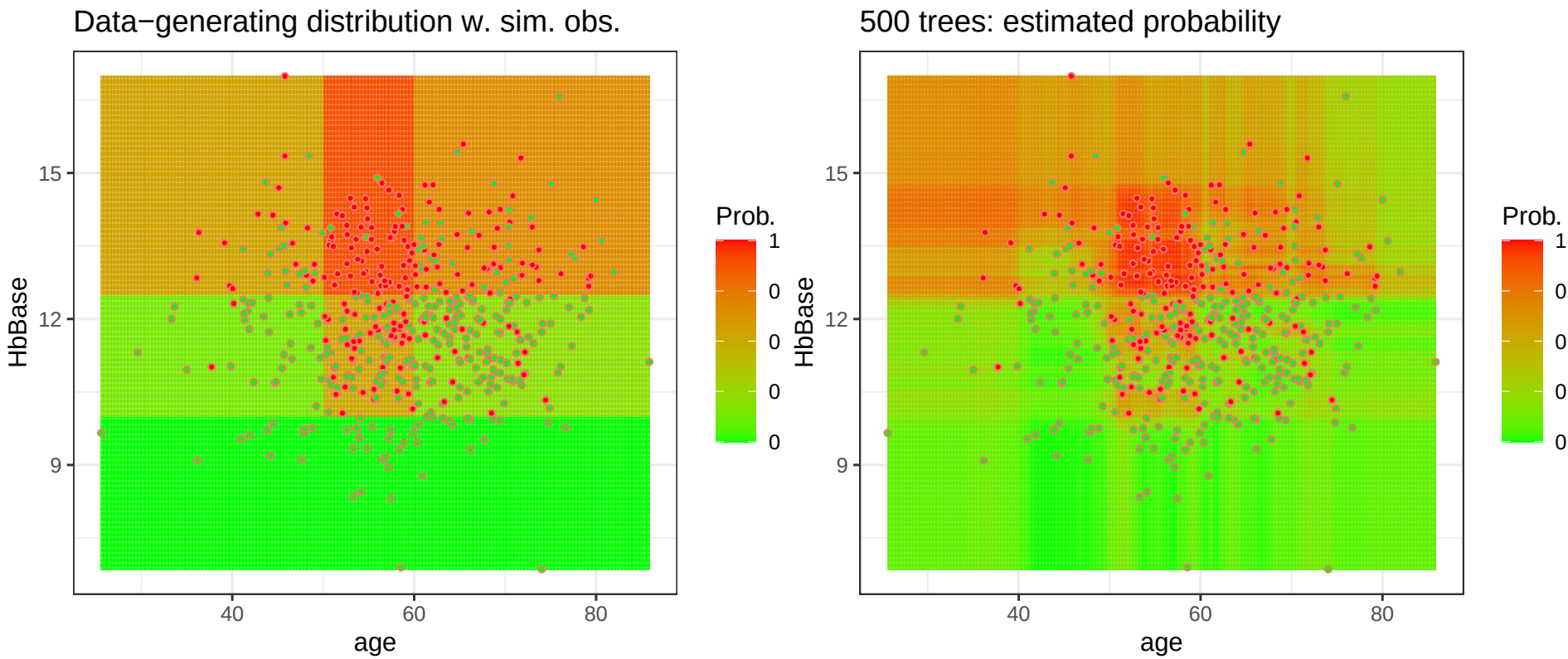
Tuning ntree, nodesize and mtry



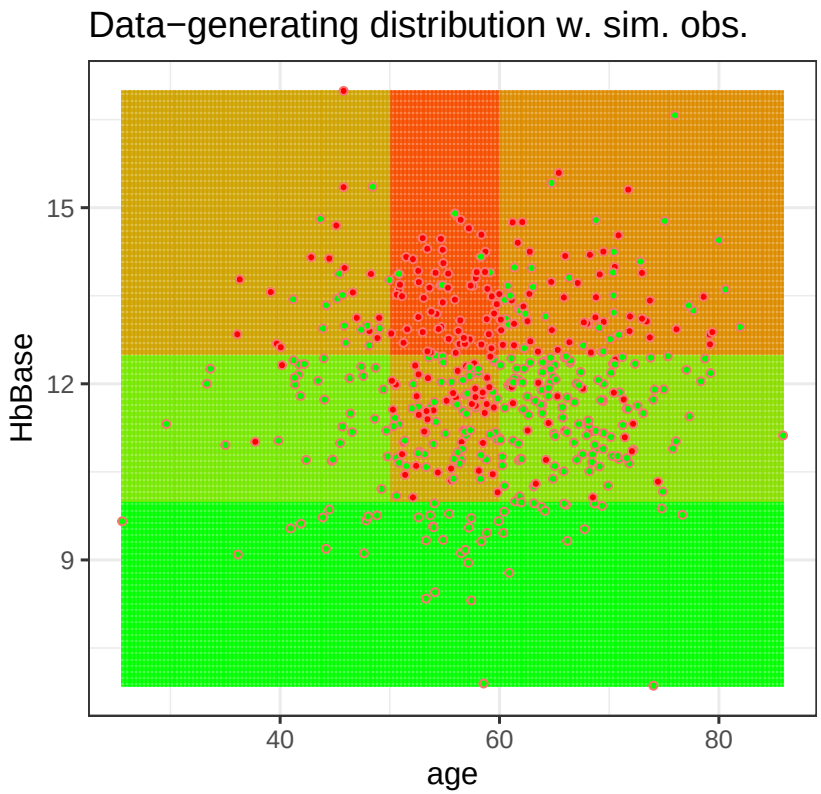
The smallest OOB error

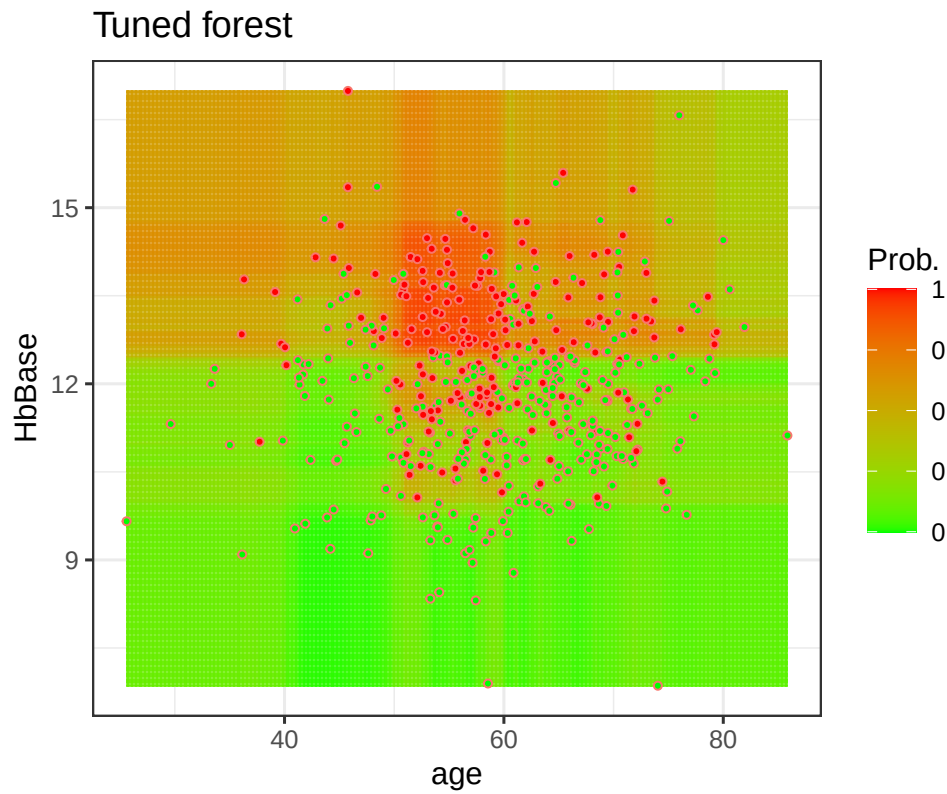
```
mtry nodesize oob.error
   2        5 0.1004156
```

Tuning hyperparameters for the simulated data



Tuning hyperparameters for the simulated data





Variable importance

When is Random Forests cool?

You want a machine to help in selecting what is important (which predictors) for making better predictions

The machine (the tuned Random Forest model) has selected how and which the variables are used to predict the probability of event.

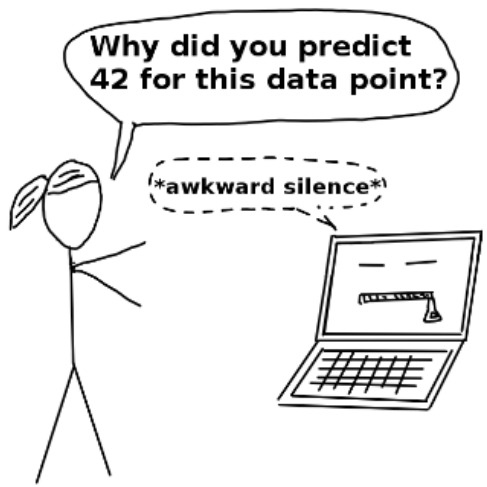
We now want to understand which variables are the most important and how they affect the prediction.

Recap: When is it not cool?

You pay by losing some of the traditional statistical inference

This includes how covariates influence the outcome (e.g. usual coefficients from e.g. linear regression models, or summary statistics such as OR, RR and HR).

Hypotheses about causality becomes untestable in the traditional statistical sense.



Epo example

Treatment with epoetin beta was defined successful when the hemoglobin level increased sufficiently. For patient i set

$$Y_i = \begin{cases} 1 & \text{treatment successful} \\ 0 & \text{treatment failed} \end{cases}$$

Age	min: 41 y, median: 59 y, max: 80 y
Gender	male: 85%, female: 15%
Baseline hemoglobin	mean: 12.03 g/dl, std: 1.45
Treatment	epo: 50%, placebo 50%
Resection	complete: 48%, incomplete: 19%, no resection: 34%
Epo receptor status	neg: 32%, pos: 68%

Logistic regression

Response: treatment successful yes/no

Factor	OddsRatio	StandardError	CI.95	pValue
(Intercept)	0.00	4.01	–	< 0.0001
Age	0.97	0.03	$[0.91; 1.03]$	0.2807
Sex:female	4.71	0.84	$[0.91; 26.02]$	0.0657
HbBase	3.25	0.27	$[1.99; 5.91]$	< 0.0001
Treatment:Epo	90.92	0.76	$[23.9; 493.41]$	< 0.0001
Resection:Incompl	1.75	0.81	$[0.36; 9.03]$	0.4924
Resection:Compl	4.14	0.69	$[1.13; 17.36]$	0.0395
Receptor:positive	5.81	0.66	$[1.72; 23.39]$	0.0076

Maybe we are not so sure about the model

In the logistic regression, we specify a model. To use the Odds Ratios, confidence intervals and p-values, that models needs to be correctly specified.

This includes having the correct form of the predictions in the model, e.g. specifying interactions and adding non-linearity.

This can be a difficult task with many predictors e.g. when using genetic data.

Circulation: Cardiovascular Quality and Outcomes
Volume 4, Issue 1, January 2011; Pages 39-45
https://doi.org/10.1161/CIRCOUTCOMES.110.939371

ORIGINAL ARTICLE

Identifying Important Risk Factors for Survival in Patient With Systolic Heart Failure Using Random Survival Forests

Eileen Hsich, MD, Eiran Z. Gorodeski, MD, MPH, Eugene H. Blackstone, MD, Hemant Ishwaran, PhD, and Michael S. Lauer, MD

BACKGROUND— Heart failure survival models typically are constructed using Cox proportional hazards regression. Regression modeling suffers from a number of limitations, including bias introduced by commonly used variable selection methods. We illustrate the value of an intuitive, robust approach to variable selection, random survival forests (RSF), in a large clinical cohort. RSF are a potentially powerful extensions of classification and regression trees, with lower variance and bias.

Identifying Important Risk Factors for Survival in Patient With Systolic Heart Failure Using Random Survival Forests

Eileen Hsich, MD; Eiran Z. Gorodeski, MD, MPH; Eugene H. Blackstone, MD; Hemant Ishwaran, PhD; Michael S. Lauer, MD

Background—Heart failure survival models typically are constructed using Cox proportional hazards regression. Regression modeling suffers from a number of limitations, including bias introduced by commonly used variable selection methods. We illustrate the value of an intuitive, robust approach to variable selection, random survival forests (RSF), in a large clinical cohort. RSF are a potentially powerful extensions of classification and regression trees, with lower variance and bias.

Methods and Results—We studied 2231 adult patients with systolic heart failure who underwent cardiopulmonary stress testing. During a mean follow-up of 5 years, 742 patients died. Thirty-nine demographic, cardiac and noncardiac comorbidity, and stress testing variables were analyzed as potential predictors of all-cause mortality. An RSF of 2000 trees was constructed, with each tree constructed on a bootstrap sample from the original cohort. The most predictive variables were defined as those near the tree trunks (averaged over the forest). The RSF identified peak oxygen consumption, serum urea nitrogen, and treadmill exercise time as the 3 most important predictors of survival. The RSF predicted survival similarly to a conventional Cox proportional hazards model (out-of-bag C-index of 0.705 for RSF versus 0.698 for Cox proportional hazards model).

Conclusions—An RSF model in a cohort of patients with heart failure performed as well as a traditional Cox proportional hazard model and may serve as a more intuitive approach for clinicians to identify important risk factors for all-cause mortality. **(*Circ Cardiovasc Qual Outcomes*. 2011;4:39-45.)**

Key Words: heart failure ■ prognosis ■ statistics ■ survival analyses

Two sides to the story

We can describe some of the behavior of the random forest by looking at two perspectives:

- Which predictors are mostly used the prediction.
- How the predictor affects the prediction.

When is a variable important?

We have used a loss function (Brier Score) so far to evaluate our models.

Variable importance can be evaluated by investigating what happens to the performance (e.g. Brier Score) of the Random Forest with and without a variable. Doing this for all variables, we can rank the variables in terms of how much error they reduce.

As before it is a good idea to use the out of bag data.

Variable Importance (VIMP)

We can use the out-of-bag predictions to calculate the variable importance. This can be useful for:

- Identifying the most important variables
- Selecting variables

We will here look specifically at permutation importance.

Permutation importance

VIMP (Variable IMPortance) is measured by the difference in prediction error between:

- running the forest with a "noised-up" version of X
- running the forest with X as was observed

If prediction performance decreases more for variable X_1 than for variable X_2, then importance$(X_1) >$ importance(X_2)

Variable Importance (VIMP)

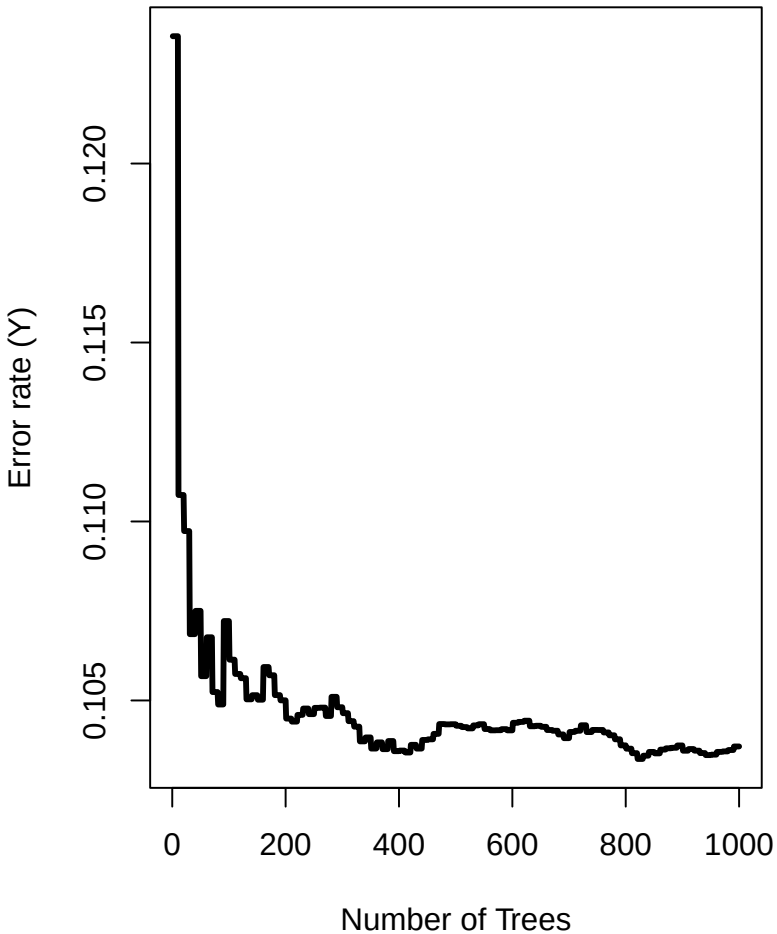
In R, we can get the values from the permutation importance by:

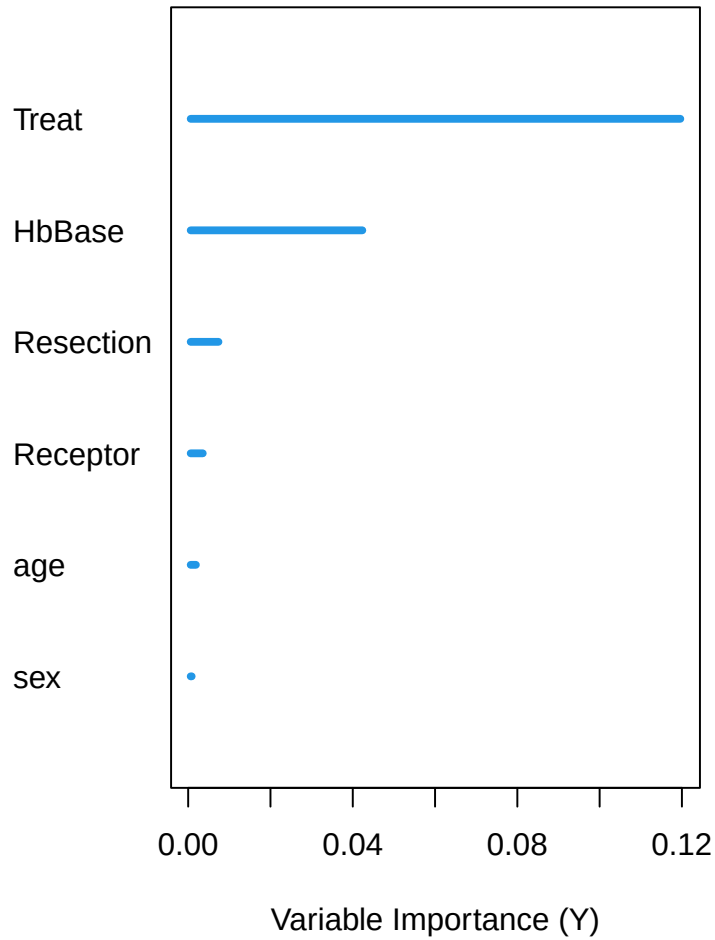
```
tuned.rf <-
  rfsrc(Y ~ age+sex+HbBase+Treat+Resection+Receptor,
    data = Epo, mtry = 2, nodesize = 5,
    ntree = 1000,
    importance = TRUE, # getting VIMP values
    seed = 6)
```

And plot them by:

```
plot(tuned.rf)
```

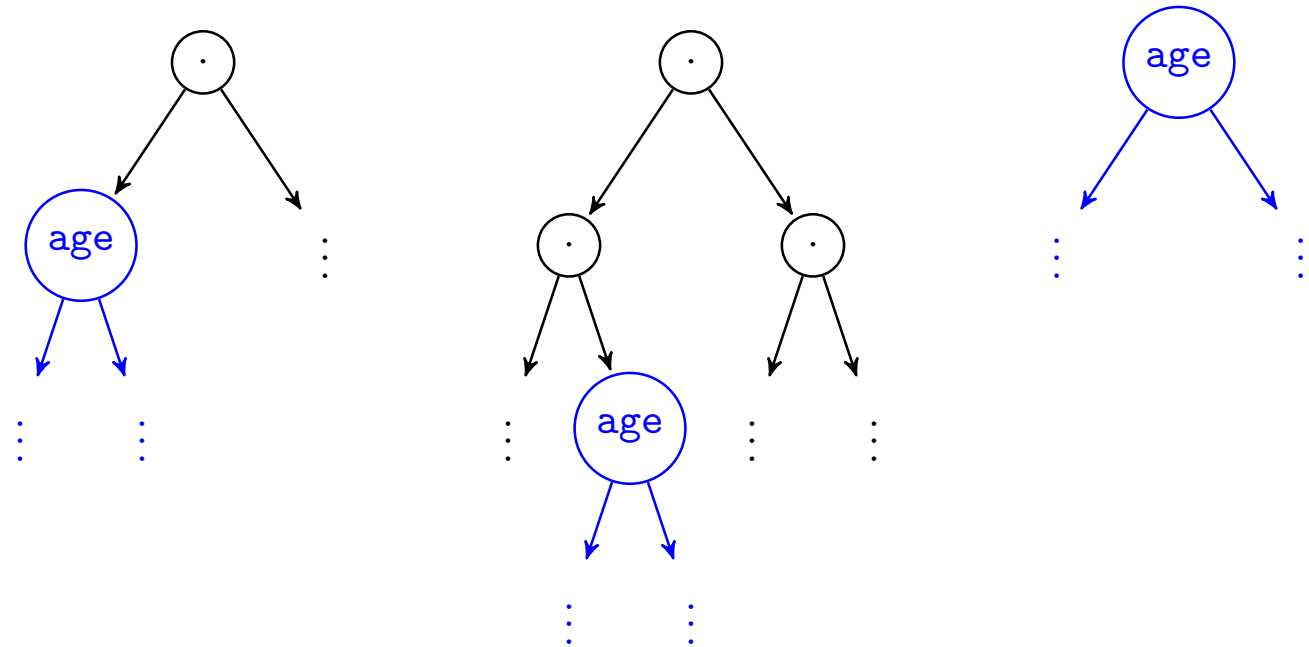
Variable Importance (VIMP)





Minimal depth

The minimal depth is the average distance from the root node to the first split on a specific variable



The smaller the minimal depth, the more important is the variable

Minimal depth

Run a random forest with `importance=TRUE`.

```
rf.imp <- rfsrc(Y~age+sex+HbBase+Treat+
                  Resection+Receptor,
                data = Epo, nodesize = 5,
                seed = 6,
                ntree = 1000,
                importance = TRUE)
```

Minimal depth

```
md <- max.subtree(rf.imp, max.order=0)
out <- sapply(1:dim(md$order)[1], function(i) mean(md$
    order[i, ]))
names(out) <- names(md$order[, 1])
print(out)
```

```
  age       sex    HbBase     Treat Resection  Receptor
1.811     3.520     1.438     1.174     2.136     2.684
```

```
md$threshold
```

```
[1] 2.404312
```

Partial Dependence Plots (PDPs)

A way to show how a predictor affects the prediction.

Partial dependent plots (PDP) gives a graphical representation of how by keeping all other predictors fixed. This is known as marginalization.

Partial Dependence Plot (PDPs)

Say, we want to know how

$$\hat{P}(Y = 1 \mid \text{age}, \text{Gender}, \text{HbBase}, \text{Treatment}, \text{Resection}, \text{Epo})$$

varies when HbBase varies

We estimate this by:

$$\hat{P}^{\text{HbBase}}(b) = \frac{1}{n} \sum_{i=1}^{n} \hat{P}(Y = 1 \mid \text{age}_i, \text{Gender}_i, \text{HbBase} = b, \text{Treatment}_i, \text{Resection}_i, \text{Epo}_i)$$

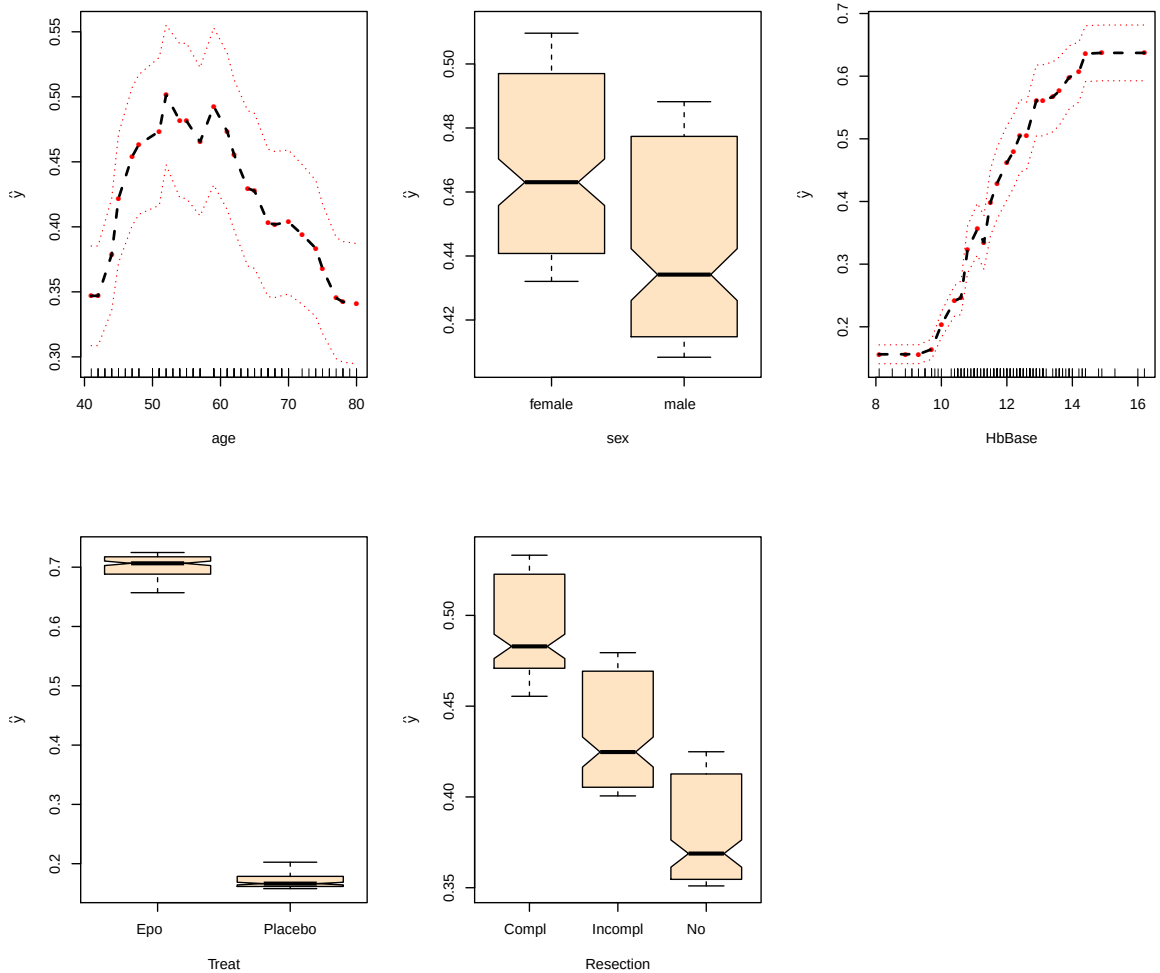
- We marginalize the forest prediction over the other features/covariates

Partial Dependence Plots (PDPs)

In R, we can plot these estimates for all variables by simply writing:

```
plot.variable(tuned.rf,
              partial=TRUE, plots.per.page=3)
```

Partial Dependence Plots (PDPs)



Other methods

Different types of predictor effect plots[9]:

Partial Dependence Plots: "Let me show you what the model predicts on average when each data instance has the value v for that feature. I ignore whether the value v makes sense for all data instances."

Other options include M-plots and ALE plots.

[9] Molnar, C. (2020). Interpretable Machine Learning.

Exercise 4: Identifying risk factors

In this exercise we consider the paper:

Identifying Important Risk Factors for Survival in Patient With Systolic Heart Failure Using Random Survival Forests

Eileen Hsich, MD; Eiran Z. Gorodeski, MD, MPH; Eugene H. Blackstone, MD;
Hemant Ishwaran, PhD; Michael S. Lauer, MD

Background—Heart failure survival models typically are constructed using Cox proportional hazards regression. Regression modeling suffers from a number of limitations, including bias introduced by commonly used variable selection methods. We illustrate the value of an intuitive, robust approach to variable selection, random survival forests (RSF), in a large clinical cohort. RSF are a potentially powerful extensions of classification and regression trees, with lower variance and bias.

Methods and Results—We studied 2231 adult patients with systolic heart failure who underwent cardiopulmonary stress testing. During a mean follow-up of 5 years, 742 patients died. Thirty-nine demographic, cardiac and noncardiac comorbidity, and stress testing variables were analyzed as potential predictors of all-cause mortality. An RSF of 2000 trees was constructed, with each tree constructed on a bootstrap sample from the original cohort. The most predictive variables were defined as those near the tree trunks (averaged over the forest). The RSF identified peak oxygen consumption, serum urea nitrogen, and treadmill exercise time as the 3 most important predictors of survival. The RSF predicted survival similarly to a conventional Cox proportional hazards model (out-of-bag C-index of 0.705 for RSF versus 0.698 for Cox proportional hazards model).

Conclusions—An RSF model in a cohort of patients with heart failure performed as well as a traditional Cox proportional hazard model and may serve as a more intuitive approach for clinicians to identify important risk factors for all-cause mortality. (***Circ Cardiovasc Qual Outcomes.*** **2011;4:39-45.**)

Key Words: heart failure ■ prognosis ■ statistics ■ survival analyses

Exercise 4: Identifying risk factors

1. Read page 39–40 of the paper
   - What is the aim of the paper?
   - What method did they use?
2. Read the Results section (p. 41f)
   - What are the results of the analysis?
   - How are the results presented? (See Figures p. 43)