# Prediction rule ensembles

## an accurate and interpretable method for prediction

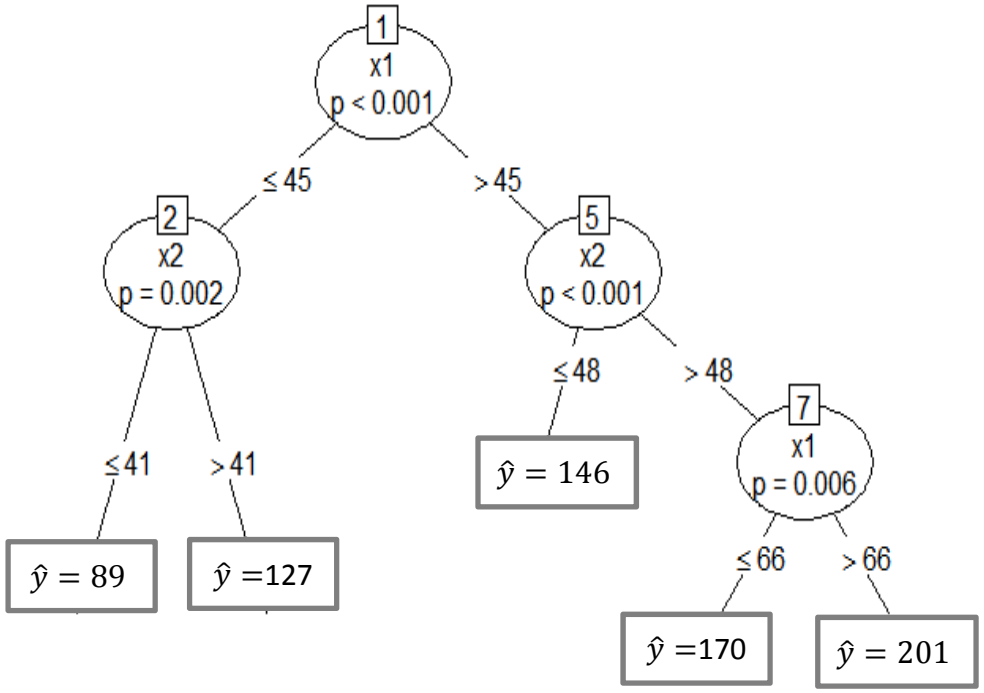
Marjolein Fokkema, Universiteit Leiden

May 2018

# Trees

Linear model: $\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots$

Decision tree:



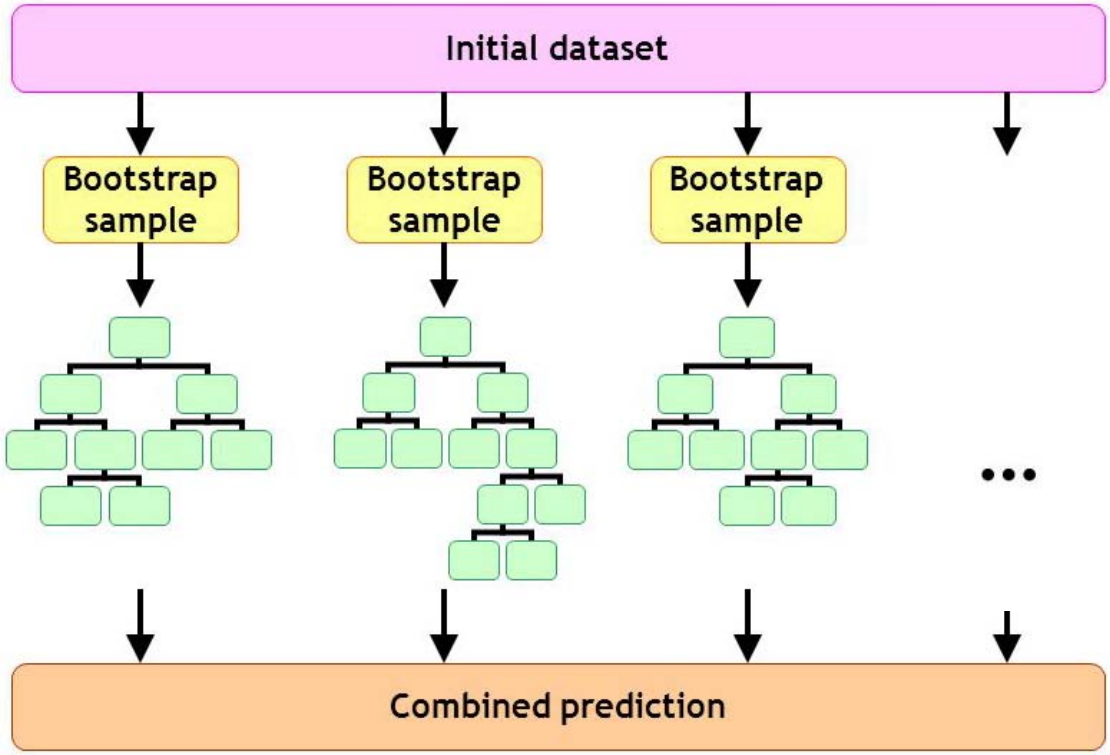
# Trees

Good: Easily interpretable and applicable

Bad: Not most accurate method

Ugly: Unstable

# Tree ensembles



# Tree ensembles

+ + + High predictive accuracy

- - - Difficult to interpret and apply

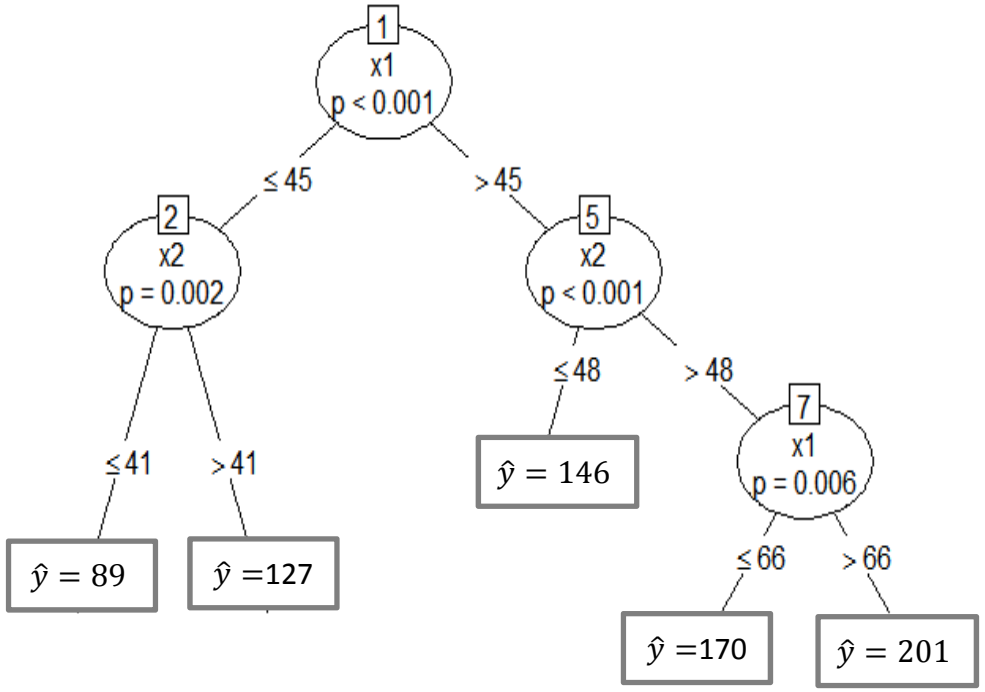
- Lots of trees
- Prediction requires lots of computation and information

Strike balance: Prediction rule ensembles

- From full tree ensemble, select only nodes that contribute most to predictive accuracy
- e.g., Rulefit (Friedman & Popescu, 2008), Node Harvest (Meinshausen, 2010)

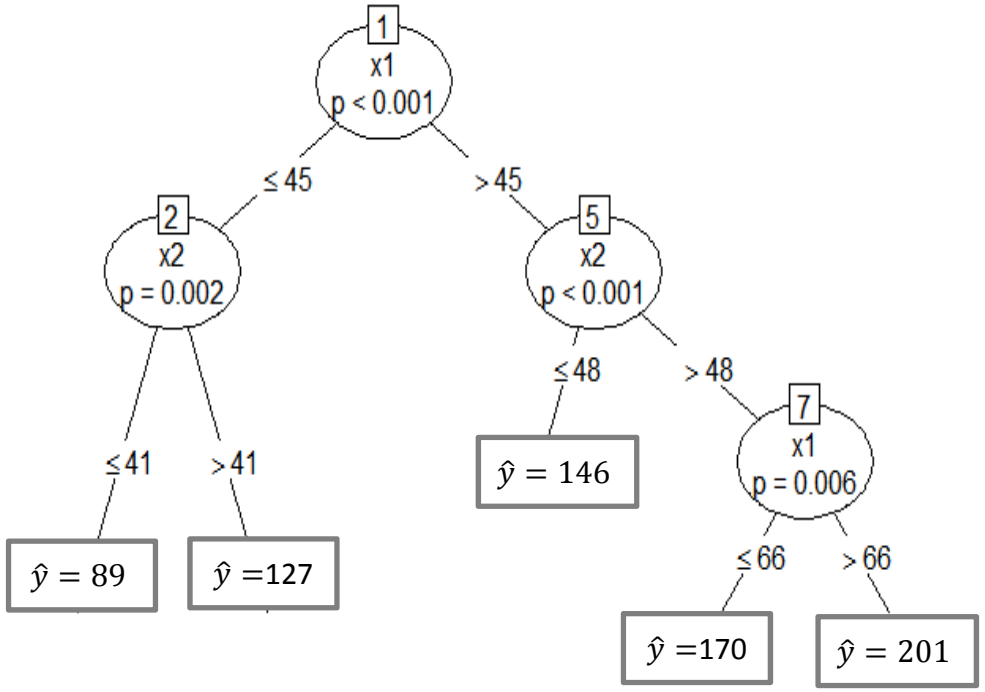
# Rule generation

$r_2(\mathrm{x}) = I(x1 \leq 45)$
$r_3(\mathrm{x}) = I(x1 \leq 45) \cdot I(x2 \leq 41)$
$r_4(\mathrm{x}) = I(x1 \leq 45) \cdot I(x2 > 41)$
$r_5(\mathrm{x}) = I(x1 > 45)$
$r_6(\mathrm{x}) = I(x1 > 45) \cdot I(x2 \leq 48)$
$r_7(\mathrm{x}) = I(x1 > 45) \cdot I(x2 > 48)$
$r_8(\mathrm{x}) = I(x1 > 66) \cdot I(x2 > 48)$
$r_9(\mathrm{x}) = I(x1 > 66) \cdot I(x2 > 48)$



# Rule generation

$r_2(\mathrm{x}) = I(x1 \leq 45)$
$r_3(\mathrm{x}) = I(x1 \leq 45) \cdot I(x2 \leq 41)$
$r_4(\mathrm{x}) = I(x1 \leq 45) \cdot I(x2 > 41)$
~~$r_5(\mathrm{x}) = I(x1 > 45)$~~
$r_6(\mathrm{x}) = I(x1 > 45) \cdot I(x2 \leq 48)$
$r_7(\mathrm{x}) = I(x1 > 45) \cdot I(x2 > 48)$
$r_8(\mathrm{x}) = I(x1 > 66) \cdot I(x2 > 48)$
$r_9(\mathrm{x}) = I(x1 > 66) \cdot I(x2 > 48)$



# Rulefit algorithm (Friedman & Popescu, 2008)

1) Take subsamples from training data
2) Grow tree on every sample
   - Boosting with CART trees
3) Create initial ensemble
   - Includes every node from every tree as a rule, and/or
   - Predictor variables as linear functions
4) Select final ensemble by sparse regression on training data
   - Lasso, ridge, elastic net, forward stepwise

Implemented in Fortran: very fast

But: not open source, not flexible, not well-documented, CART has variable selection bias, supports only binary and continuous outcomes

# R package **pre**
# (Fokkema & Christoffersen, 2017)

1) Take sub- or bootstrap samples from training data
2) Grow tree on every sample

   Tuning parameters: partitioning algorithm (ctree, glmtree or CART), #trees, mtry, sampling fraction, tree depth, learning rate, …
3) Create initial ensemble
   - Every node from every tree as a rule, and/or
   - Predictor variables as linear functions
   - Experimental: multivariate adaptive regression splines, (user-defined) base learners
4) Select final ensemble by sparse regression on training data

   Currently only lasso, ridge or elastic net

# Example: Depression data

Study of Carrillo et al. (2001), N = 112

Response: BDI (Beck Depression Inventory)

Potential predictors:

- Personality scales:
  - Neuroticism: n1, n2, n3, n4, n5, n6, ntot
  - Extraversion: e1, e2, e3, e4, e5, e6, etot
  - Openness: open1, open2, open3, open4, open5, open6, opentot
  - Altruism: altot
  - Conscientiousness: contot
- Sex
- Age in years

# Example: Depression data

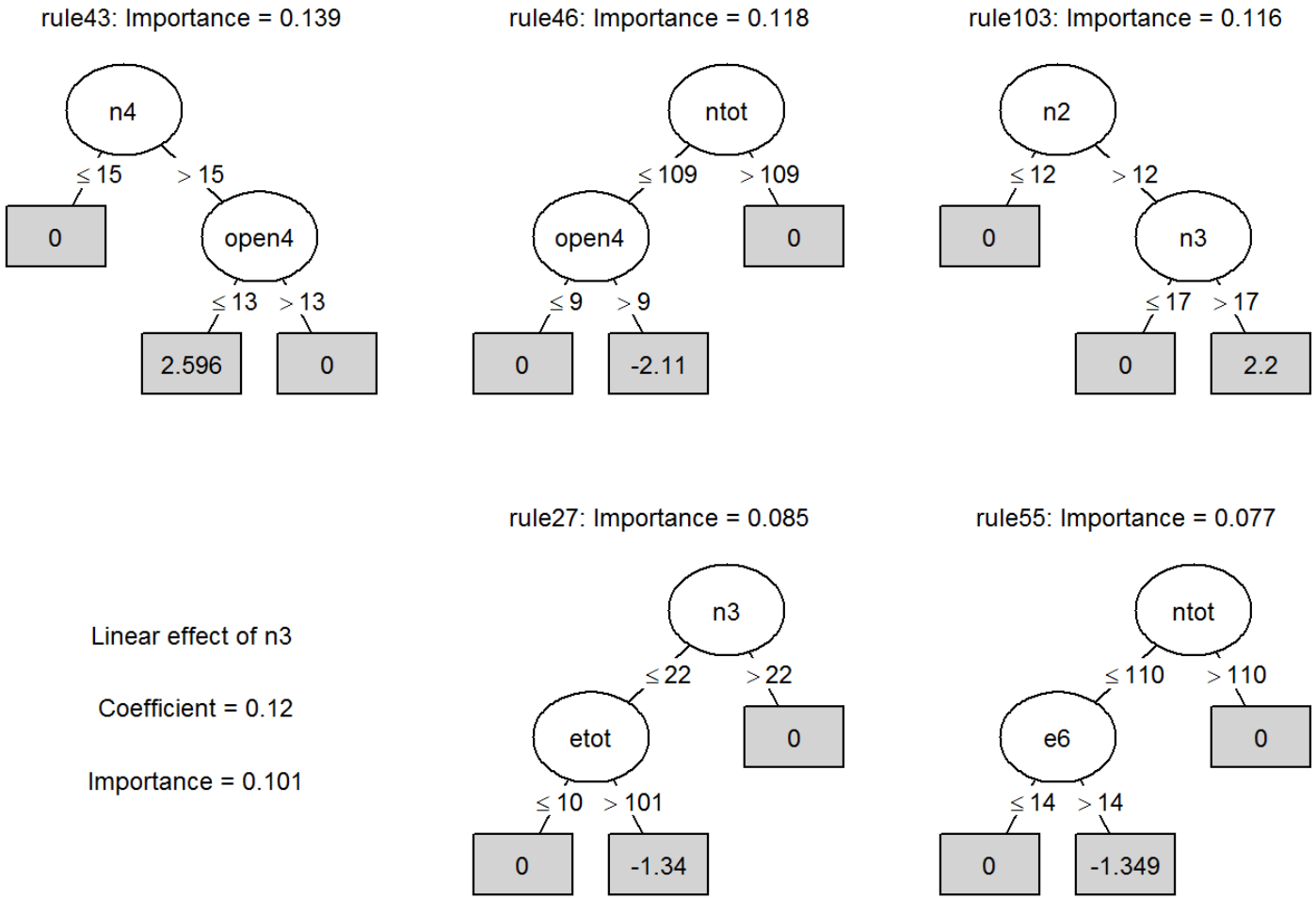
```
library("pre")
data("carrillo")
```

```
set.seed(3432)
bdi.ens <- pre(bdi ~ . , data = carrillo)
bdi.ens
```

```
## Final ensemble with cv error within 1se of minimum:
##   lambda =  0.6617113
##   number of terms = 9
##   mean cv error (se) = 33.36587 (6.543242)
##
##   cv error type : Mean-Squared Error
##
##          rule  coefficient              description
##   (Intercept)    9.1309834                     <NA>
##         rule43    2.6813625   n4 > 15 & open4 <= 13
##        rule103    2.2515172       n2 > 12 & n3 > 17
##         rule46   -2.1439376 ntot <= 109 & open4 > 9
##         rule27   -1.4022718   n3 <= 22 & etot > 101
##         rule55   -1.3503107   ntot <= 110 & e6 > 14
##         rule57   -0.5293091   n3 <= 17 & open4 > 10
##         rule68   -0.3507426                 n1 <= 20
##         rule81   -0.2458626                 n1 <= 23
##             n3    0.1225455       2 <= n3 <= 30.225
```

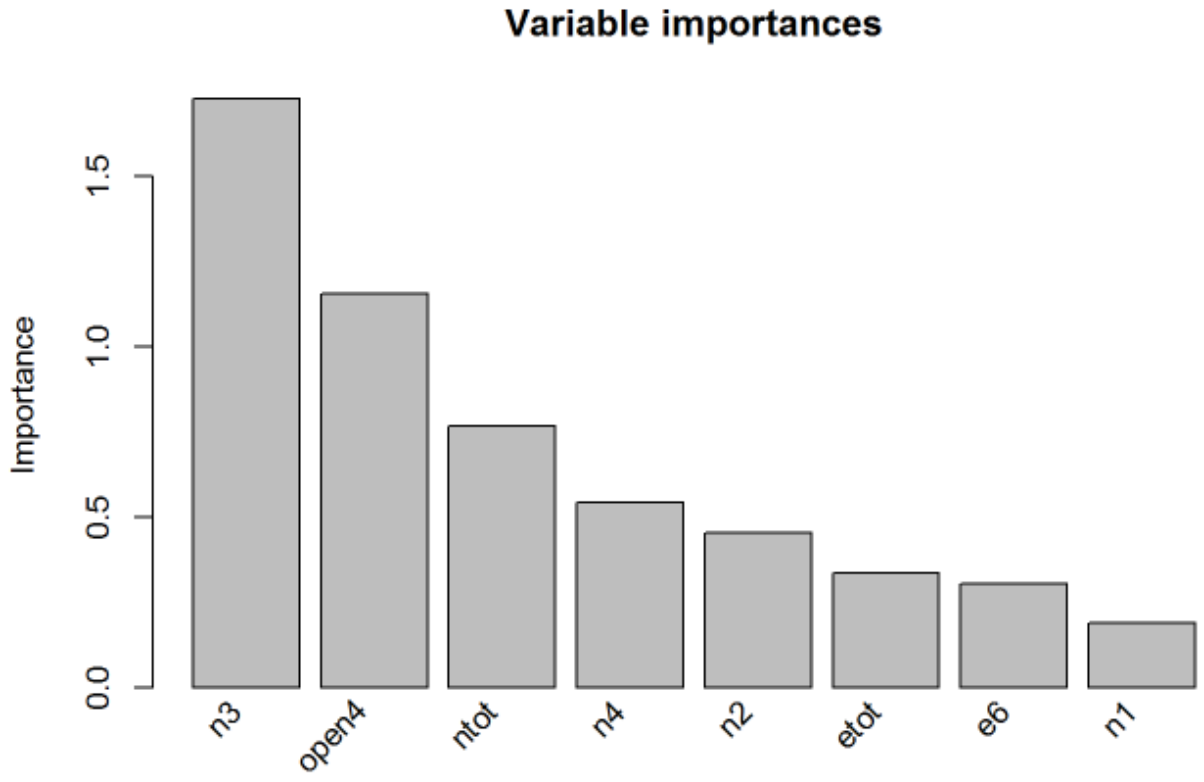
# Example: Depression data

```
plot(bdi.ens)
```



# Example: Depression data

```
imps <- importance(bdi.ens, round = 4)
```



# Example: Depression data

```
imps
```

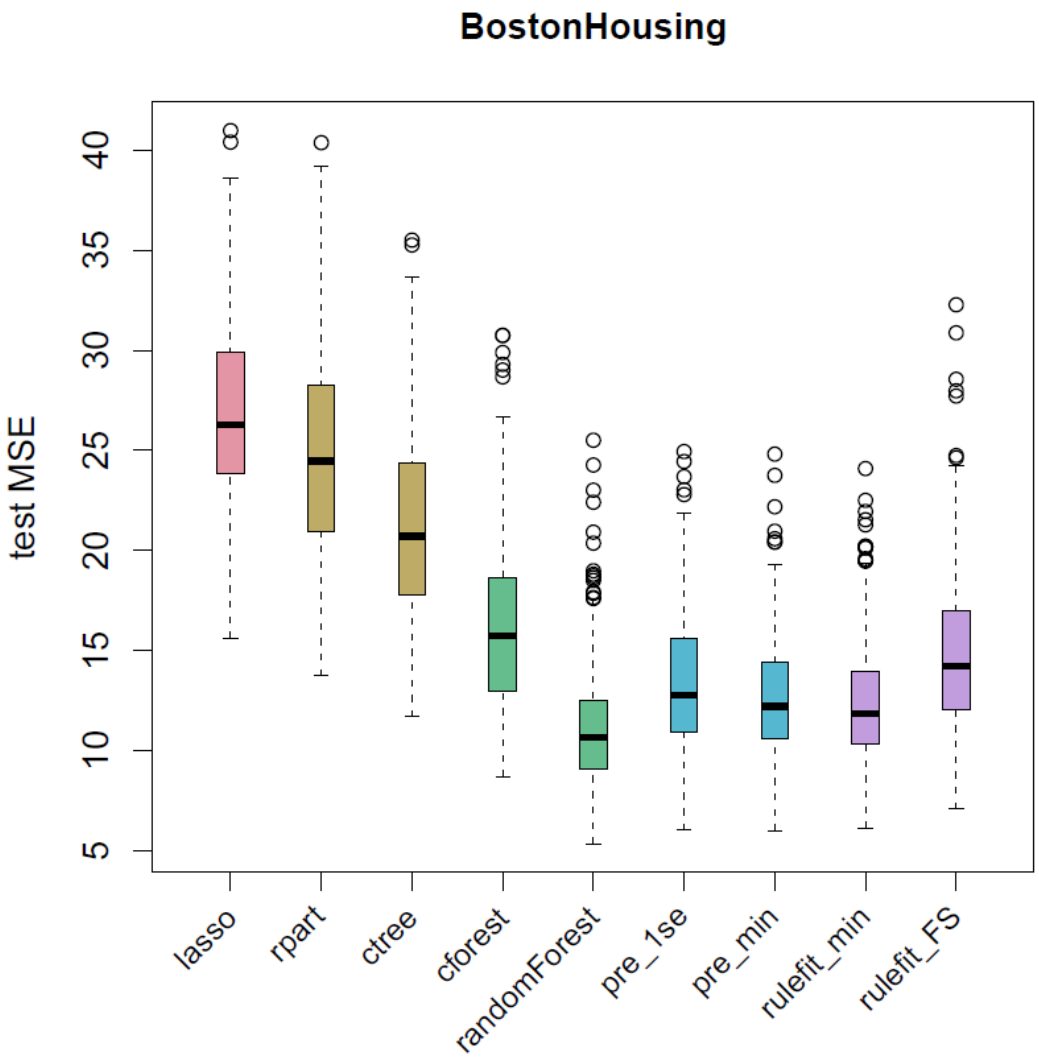
```
## $baseimps
##      rule                description    imp coefficient     sd
## 1  rule43   n4 > 15 & open4 <= 13 1.1215      2.6814 0.4183
## 2  rule46 ntot <= 109 & open4 > 9 0.9434     -2.1439 0.4400
## 3 rule103       n2 > 12 & n3 > 17 0.9280      2.2515 0.4122
## 4      n3       2 <= n3 <= 30.225 0.8092      0.1225 6.6030
## 5  rule27   n3 <= 22 & etot > 101 0.7002     -1.4023 0.4994
## 6  rule55   ntot <= 110 & e6 > 14 0.6069     -1.3503 0.4494
## 7  rule57   n3 <= 17 & open4 > 10 0.2652     -0.5293 0.5010
## 8  rule68                n1 <= 20 0.1678     -0.3507 0.4785
## 9  rule81                n1 <= 23 0.0841     -0.2459 0.3421
```

# Resolution

Does pre kill the bad?

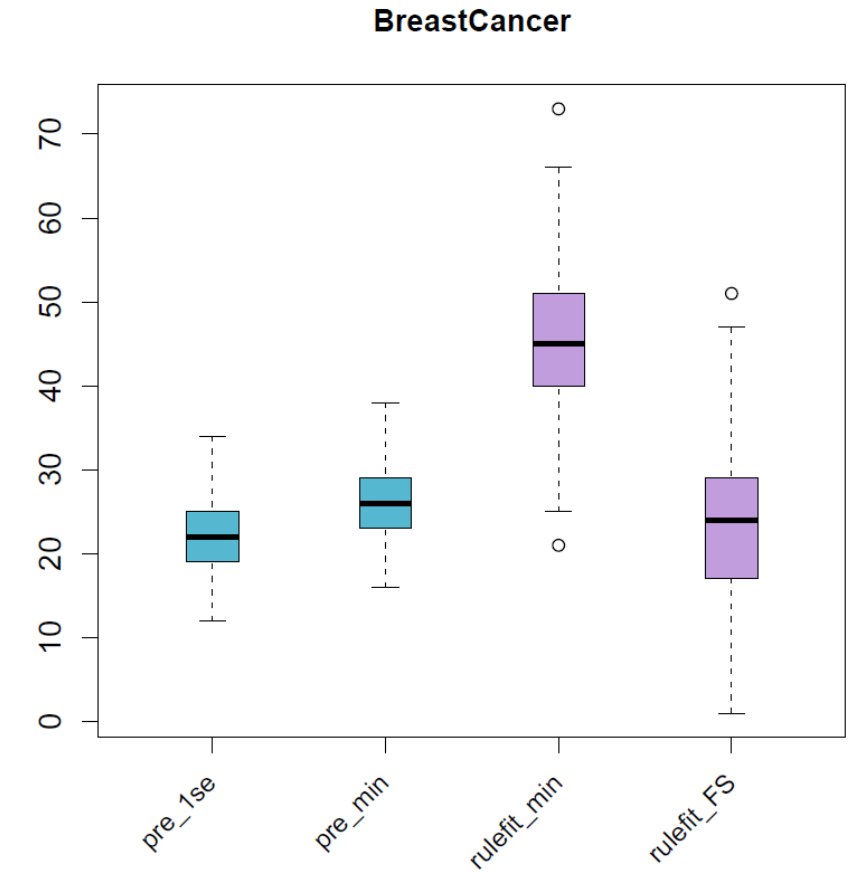
Does the good survive?

# Predictive accuracy



# Complexity


BostonHousing
# of variables selected
6
8
10
12
It's just a flesh wound.


BreastCancer
# of terms selected
0
10
20
30
40
50
60
70
pre_1se
pre_min
rulefit_min
rulefit_FS

# Discussion

- **pre** provides accuracy competitive with random forests and original RuleFit, with better interpretability
  - Lower complexity mostly due to employing unbiased recursive partitioning instead of traditional CART
- Current and future topics:
  - Modeling of clustered and longitudinal data
    - Incorporating random effects estimation dramatically increases computation time, but may improve predictive accuracy
  - Enforcing user-specified sparsity
    - E.g., how to select and determine weights if I want just five rules, not data-driven selection of best number of rules

# Thank you for your attention!

**References**

Carrillo, J. M., Rojo, N., Sanchez-Bernardos, M. L., & Avia, M. D. (2001). Openness to experience and depression. *European Journal of Psychological Assessment, 17*(2), 130.

Fokkema, M. (2017). pre: An R Package for Fitting Prediction Rule Ensembles. Working paper, arXiv:1707.07149.

Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics, 916-954.*

Meinshausen, N. (2010). Node harvest. *The Annals of Applied Statistics 4*(4), 2049-2072.

Package **pre** is available from:

https://CRAN.R-project.org/package=pre

https://github.com/marjoleinF/pre

m.fokkema@fsw.leidenuniv.nl

# Clustered data structures

Final ensemble (additive model):

Rule, e.g.:
$f_m(\mathbf{x}) = I(x_1 > 5) \cdot I(x_4 \leq 10)$

$$F(\mathbf{x}) = \hat{a}_0 + \sum_{m=1}^{M} \hat{a}_m f_m(\mathbf{x})$$

Linear term, e.g.:
$f_m(\mathbf{x}) = x_2$

What if data are clustered?

-> Ignore clustered structure

-> Sample level-2 instead of level-1 units in generating rules

-> Introduce random effects into additive model