Searching for Dusty Corners: Understanding the Prediction of the Cross Section of Returns

Carlos Carvalho, Juhani Linnainmaa, Rob McCulloch

1. Goals

Predict

Simple approach to predicting the monthly cross section of firm returns using characteristics.

Use ensembles of trees, bagged through time to learn $\hat{f}(x)$

Interpret (fit the fit)

Summarize $E(R) = \hat{f}(x)$ by searching for simple fits of the fit.

Focus on **variable selection** *and* **interactions**: Can I find an additive function of a subset of the variables that approximates $\hat{f}(x)$ well.

Dusty Corners:

We think there are small parts of the predictor space where "interesting" nonlinearities kick in.

We will try to indentify variables that contribute to nonlinearity and interactions in the dusty corners.

Data:

- 629 months of data, 1963-06 - 2015-12.
- Each month we have a cross section of firm returns, and 33 firm characteristics measured in the previous month.
- threw out "tinies"
- on a monthly basis express each x as a quantile in $(0, 1)$.
- regression impute missing values
- monthly demean returns, so we are predicting amount above average

Some Key Predictor Variables

Our variable selection results will lead us to focus on these 10.

me:
market equity. "small stocks tend to earn higher average returns than big stocks."

r1_1:
prior one month return. "short term reversals".

r12_2:
prior one year return, skipping a month. "momentum effect".

industrymom (imom):
industry momentum, prior six month's return on the stock's industry.

seasonality (seas):
Stock's average return over the prior 20 years in the same month.

idiosyncraticvol (ivol):
idiosyncratic volatility. volatility of residual from three-factor model, estimated using one month of daily data.

an_booktomarket (btm):
"value effect".

an_assetgrowth (AaGr):
percentage year-to-year growth in total assets.

an_cbprofitability" (AcbProf):
Cash-based operating profitability.

ln_turn:
number of shares traded divided by the number of shares outstanding in the previous month.
A high value means there is a lot of trading activity.

R_t: cross section of returns, month t.
x_t: predictor variables used for R_t (measured at time $t-1$).

Approach:

Our overall approach is the following:

- For each month t fit a model giving $\hat{R} = \hat{f}_t(x)$.
- Roll the fitted models: $\hat{f}_t^R(x) = \sum_{j=1}^{\nu} w_j \, \hat{f}_{t-j}(x)$.
- Check that $\hat{f}_t^R(x)$ has reasonable predictive performance.
- Inspect $\{\hat{f}_t^R\}$ to learn about the relationship, (e.g., what variables are used).
- Also consider $\hat{f}^A(x) = \frac{1}{N} \sum_{t=1}^{N} \hat{f}_t(x)$.

For example, we often use $\nu = 120$, $w_j = 1/120$.

Choice of "Learner"

We have to fit a model each month so we want to use approaches that do not require a lot of tuning. In addition, our x variables are "messy" so we need methods that perform well in this case.

We focus on methods based on trees and ensembles of trees:

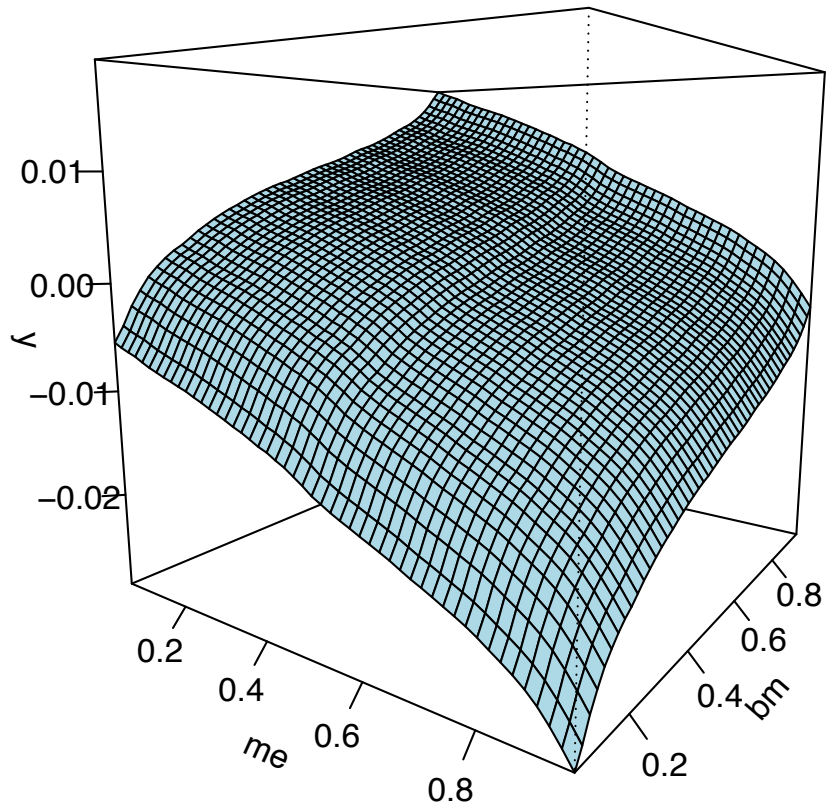
- Trees are capable of uncovering any kind of non-linearity and interaction.
- Trees handle messy x variables: they are invariant to monotonic transformations of the predictor variables.
- Single trees partition the x space into rectangular subsets somewhat reminiscent of what you obtain by sorting stocks into portfolios
- Ensembles of trees, in which many trees are combined to get an overall fit, are the best "off-the-shelf" models.
- We will use Random Forests and BART (Bayesian Additive Regression Trees) which is an ensemble method related to boosting. Generally, BART requires less tuning than other boosting type approaches. Random Forests is well known for performing well with minimal tuning.

We ran default BART and default random forests.

Our goal is to have some understanding of what the non-linear fitted relationship is.

E(R) vs
x1: me = market equity
x2: bm =book-to-market.

Hard in high dimensions!!!



Different Learners, Similar Results

Most of the of the methods could be used with estimates of $E(R \mid x)$ from any learner.

For example, Gu, Kelly and Xiu (2019) have some interesting results with neural nets.
Our results achieve a very similar accuracy level

Most of our results just examine the fit $E(R \mid x)$, but we are working on capturing the uncertainty.

2. Predictability

Is there any predictive ability?

Are the Machine Learners any better than linear?

Stacked Correlations

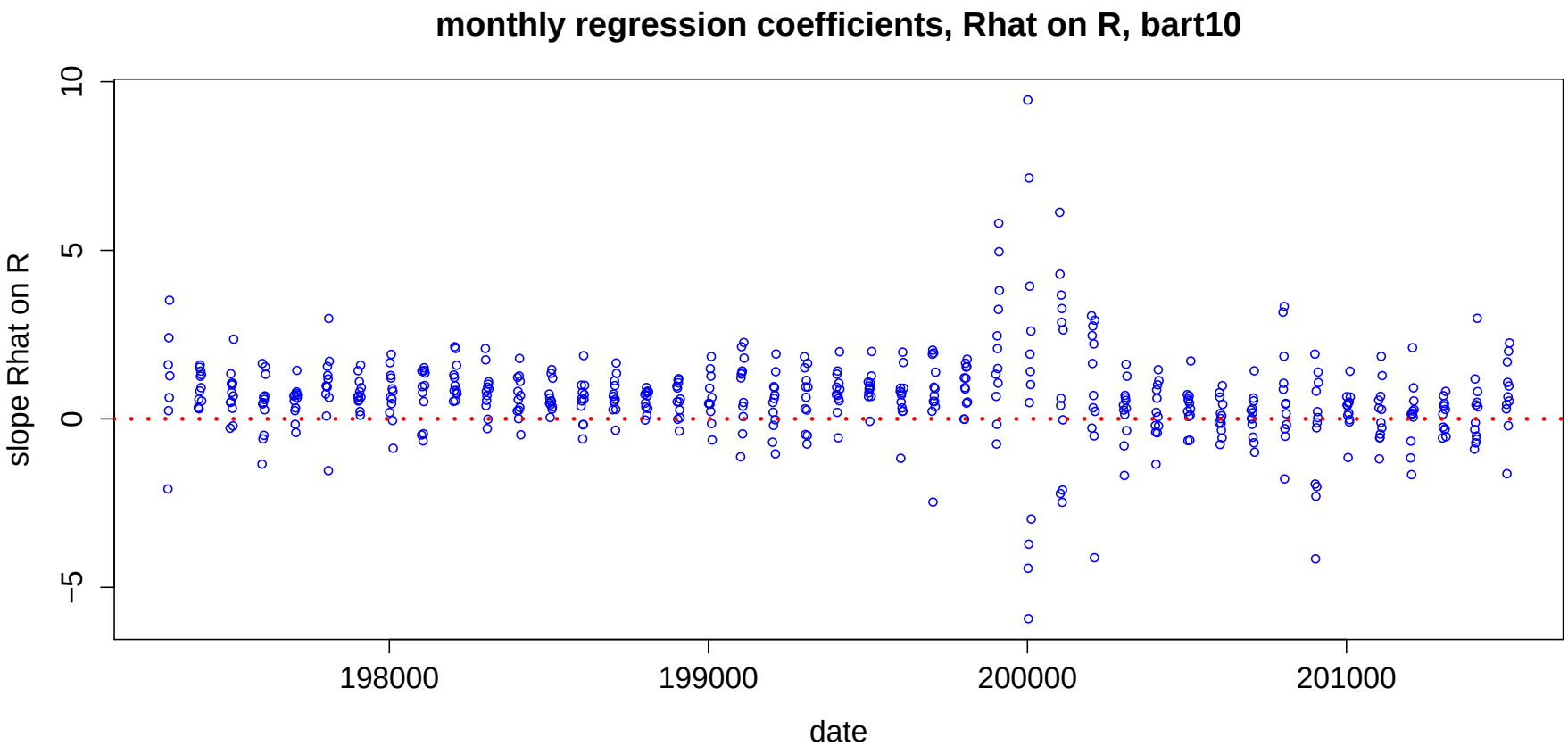
Stack all the R for each month and all the out-of-sample $\hat{R}$ for each month and compute the simple pearson correlations.

rf is Random Forests.
abart uses the average $\hat{f}$ from all months.
*10 uses just 10 variables we got from our variable selection.
tree used 25 bottom nodes.

	R	linear	tree	rf	bart	bart10	abart	abart10
R	1.0000	0.0482	0.0409	0.0468	0.0553	0.0572	0.0706	0.0693
linear	0.0482	1.0000	0.5929	0.7160	0.7993	0.7589	0.7850	0.7536
tree	0.0409	0.5929	1.0000	0.7288	0.6414	0.6278	0.5613	0.5565
rf	0.0468	0.7160	0.7288	1.0000	0.7611	0.7147	0.6580	0.6380
bart	0.0553	0.7993	0.6414	0.7611	1.0000	0.8565	0.8338	0.7825
bart10	0.0572	0.7589	0.6278	0.7147	0.8565	1.0000	0.7913	0.8505
abart	0.0706	0.7850	0.5613	0.6580	0.8338	0.7913	1.0000	0.9297
abart10	0.0693	0.7536	0.5565	0.6380	0.7825	0.8505	0.9297	1.0000

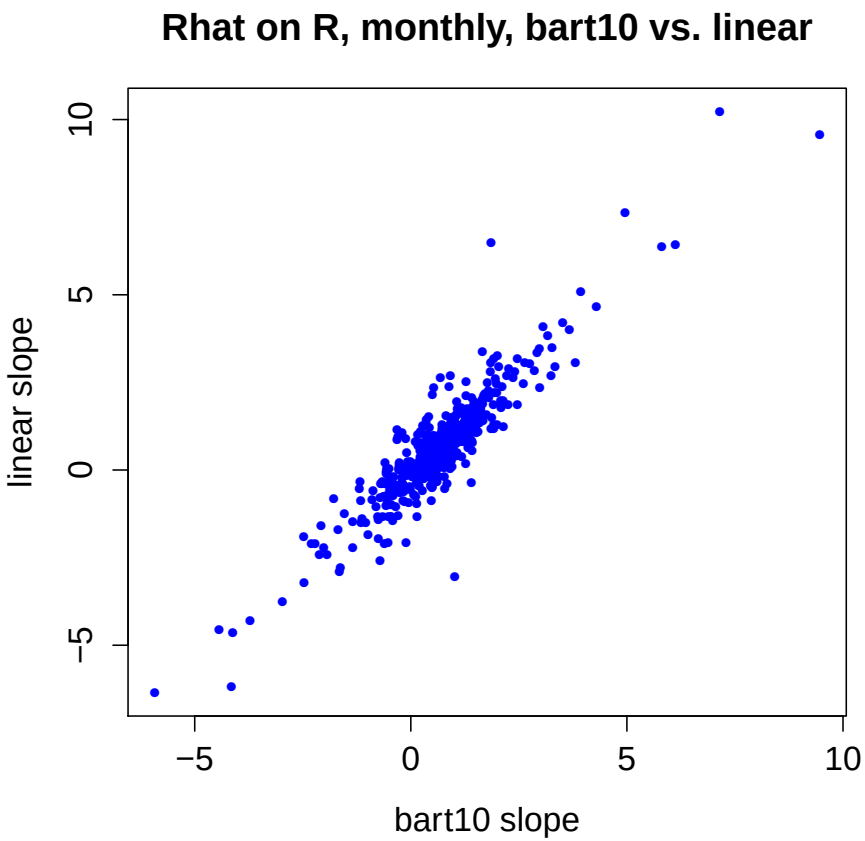

bart
Out of Sample Predictions: bart vs. linear
0.050
0.025
0.000
−0.025
−0.050
linear
−0.04
0.00
0.04
bart
count
600
400
200

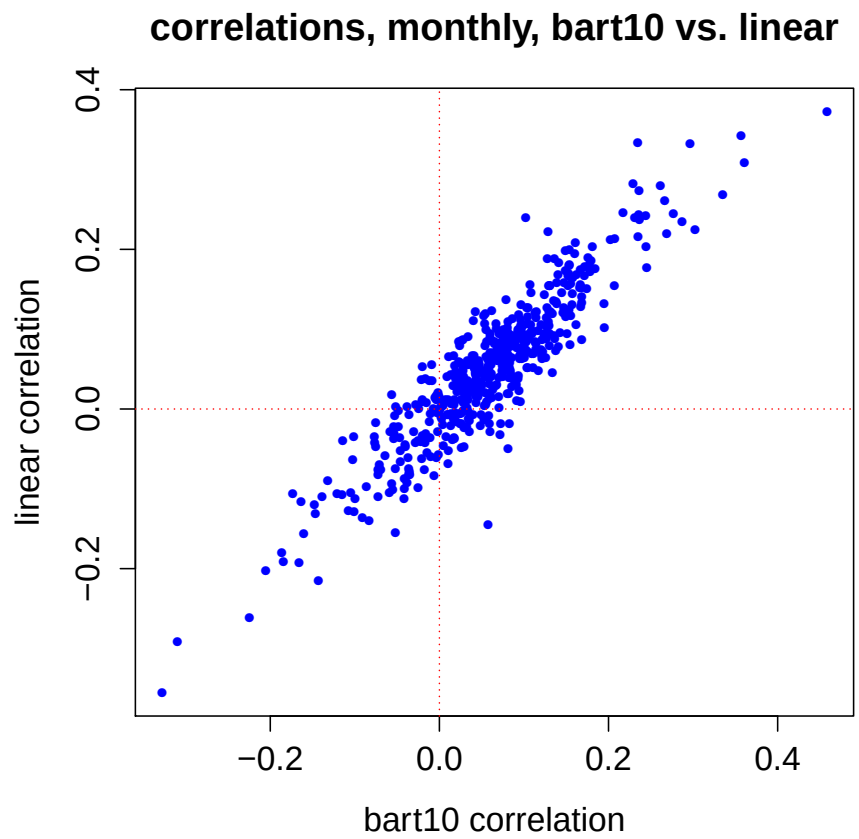
Regress out-of-sample $\hat{R}$ on R each month



What happened in 2000 ????

Compare bart10 and linear, slopes and correlations, each month





- some predictability
- *But when is the nonlinear fit different?*
- *Where are the dusty corners???*

3. Variable Selection

A key issue is

what are the important predictors ??

Tree based methods have a set tools for variable selection but we think they are all flawed.

We will use Carvalho, Hahn, McCulloch:

Fitting the fit:
variable selection using surrogate models and decision analysis

Let X^f be the set of all x of interest, $\hat{f}(X) = \{\hat{f}(x), x \in X^f\}$.

CHM assume that $\hat{f}$ is essentially the true function and then look for an approximate function

$$\gamma_S(X) \approx \hat{f}(X),$$

where $\gamma_S(X)$ uses a subset S of the predictor variables.

The key insight here is to summarize the signal once the noise is extracted

Approximating the Fit with Functions Using a Subset of the Variables:

Let $|S|$ be the size of the set S (number of variables in our case).

For each $j = 1, 2, \ldots, p-1$:

$$\underset{\gamma_S,\, |S|=j}{\text{minimize}} \; ||\hat{f}(X^f) - \gamma_S(X^f)||^2,$$

where (of course),

$$||\hat{f}(X^f) - \gamma_S(X^f)||^2 = \sum_{x \in X^f} (\hat{f}(x) - \gamma_S(x))^2.$$

For each j, we need a subset S of j variables and an approximating function γ_S using only those variables.

Remember, we don't want to make assumptions about f and hence γ_S.

We can't solve this so, as usual, we approximate our problem with a computationally feasible strategy:

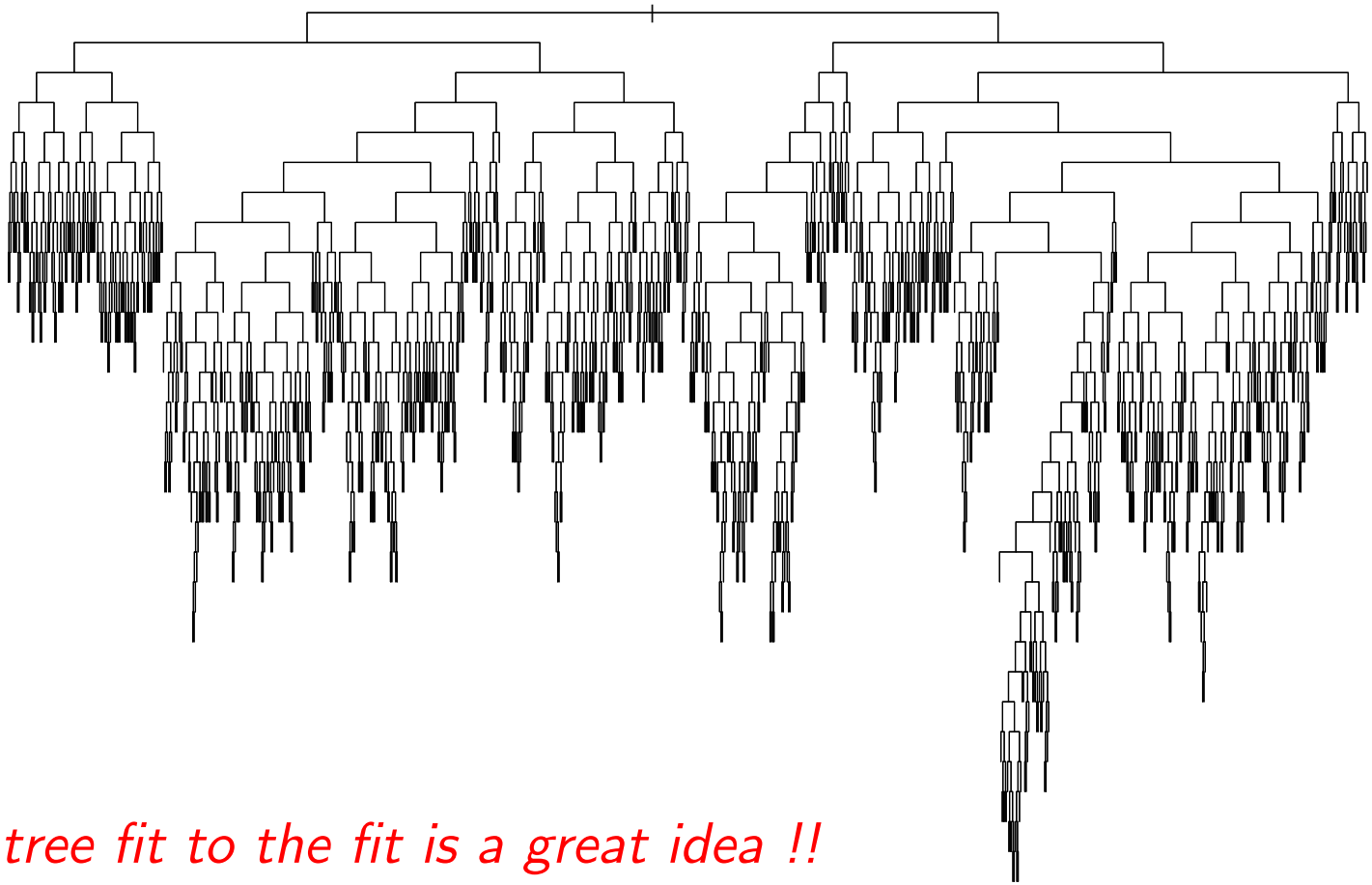
(1):

Use backwards and forwards selection to search for subsets.
As in the linear case, can do all subsets for moderate p.

(2):

Rather than run our nonparametric method (e.g. BART) using subsets of the x variables to get $\gamma_S(X^f)$, fit a big tree to $\hat{f}(X^f)$ using subsets of the x variables.

A big tree fit to the data is a terrible idea (unless you bag).



A big tree fit to the fit is a great idea !!
Forwards selection on the fit is a great idea !!
and *it is pretty fast !!!!*

We use CHM two ways:

I:

Let X be all x over all months and assets, let $\hat{f}$ be $\hat{f}^A$.

That is, use the overall average $\hat{f}$ and all the x's.

II:

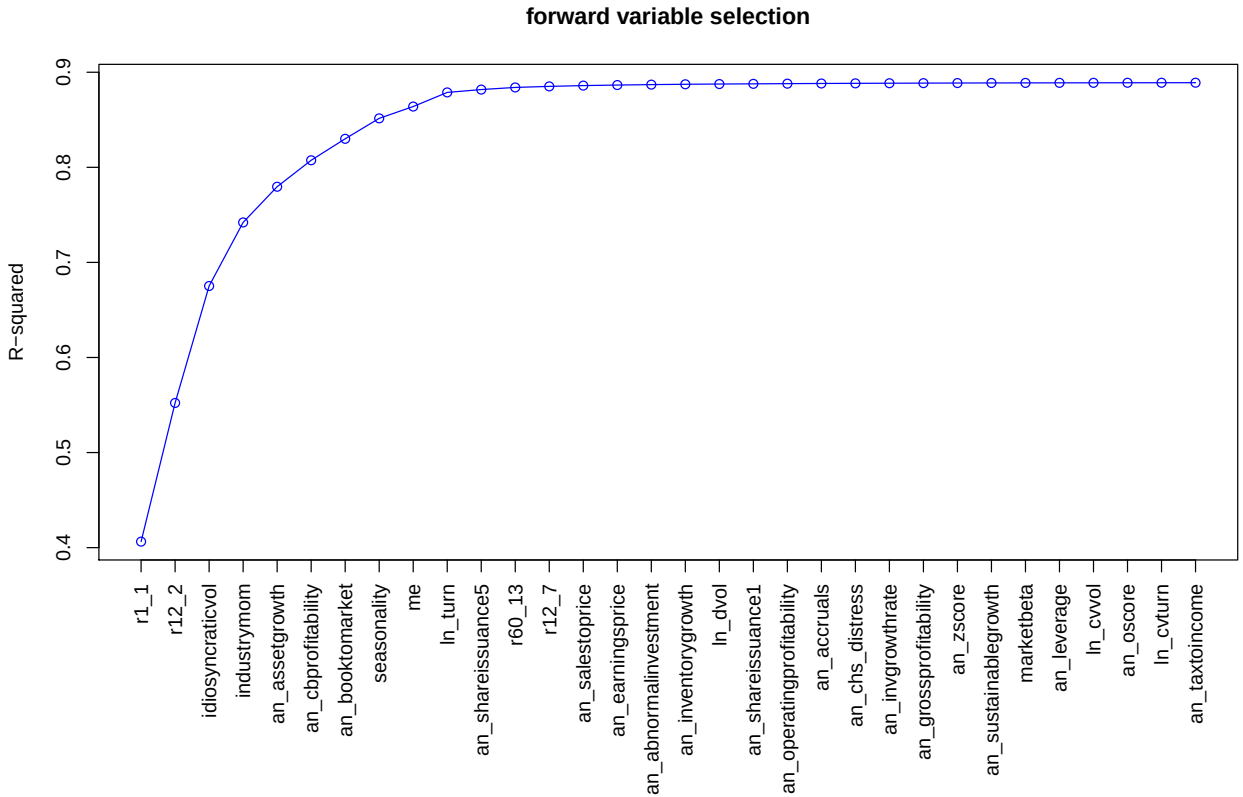
Do the variable selection for each month.

$X_t = \{x_{it}\}$, $\hat{f}_t = \hat{f}_t^R$ for each month t.

I. Results using $\hat{f}^A$

The value on the x-axis is the number of variables in S.
The value reported on the y-axis is:

$$\text{R-squared} = \text{cor}(\hat{f}^A(X), \gamma_S(X))^2.$$



As we introduce variables, going left to right, our ability to reproduce the fit using all the variables improves. After about 10 variables, there is no improvement. The results from the forward and backward searches are very similar.

Forward and Backward Variables

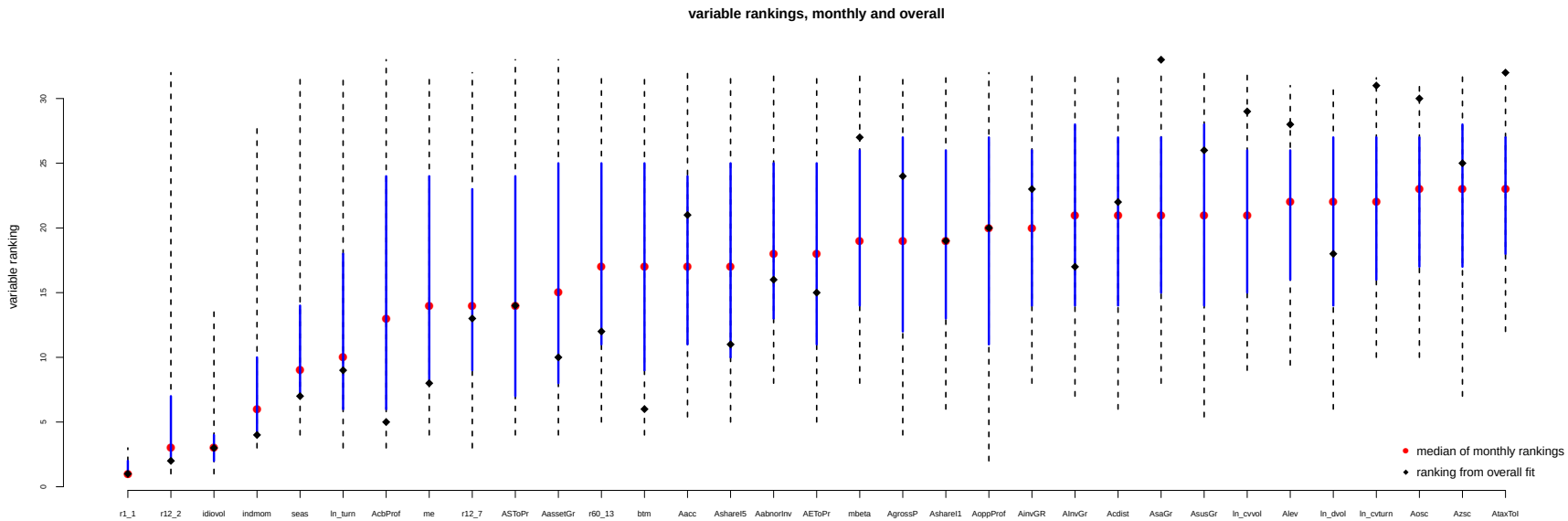
Here are the variables listed in order. So r1_1 was first in with forwards and last left with backwards.

From 10 variables on we have the same results from forward and backward search.

	namesforward	namesbackward
[1,]	"r1_1"	"r1_1"
[2,]	"r12_2"	"r12_2"
[3,]	"idiosyncraticvol"	"idiosyncraticvol"
[4,]	"industrymom"	"industrymom"
[5,]	"an_assetgrowth"	"an_cbprofitability"
[6,]	"an_cbprofitability"	"an_booktomarket"
[7,]	"an_booktomarket"	"seasonality"
[8,]	"seasonality"	"me"
[9,]	"me"	"ln_turn"
[10,]	"ln_turn"	"an_assetgrowth"
[11,]	"an_shareissuance5"	"an_shareissuance5"
[12,]	"r60_13"	"r60_13"
[13,]	"r12_7"	"r12_7"
[14,]	"an_salestoprice"	"an_salestoprice"
[15,]	"an_earningsprice"	"an_earningsprice"
[16,]	"an_abnormalinvestment"	"an_abnormalinvestment"
[17,]	"an_inventorygrowth"	"an_inventorygrowth"
[18,]	"ln_dvol"	"ln_dvol"
[19,]	"an_shareissuance1"	"an_shareissuance1"
[20,]	"an_operatingprofitability"	"an_operatingprofitability"
[21,]	"an_accruals"	"an_accruals"
[22,]	"an_chs_distress"	"an_chs_distress"
[23,]	"an_invgrowthrate"	"an_invgrowthrate"
[24,]	"an_grossprofitability"	"an_grossprofitability"
[25,]	"an_zscore"	"an_zscore"
[26,]	"an_sustainablegrowth"	"an_sustainablegrowth"
[27,]	"marketbeta"	"marketbeta"
[28,]	"an_leverage"	"an_leverage"
[29,]	"ln_cvvol"	"ln_cvvol"
[30,]	"an_oscore"	"an_oscore"
[31,]	"ln_cvturn"	"ln_cvturn"
[32,]	"an_taxtoincome"	"an_taxtoincome"
[33,]	"an_salesgrowth"	"an_salesgrowth"

Rolled Variable Selection

Here are the monthly and overall rankings for each variable.
Again, blue is 50% of months, black is 90%.
Red dot is the median over months, and black triangle is from the overall fit.



Top 10 - 15 variables rank is similar.

Bart10 is:

```
> print(colnames(TrxI)[vl10])
 [1] "me"                 "r1_1"               "r12_2"
 [4] "industrymom"        "seasonality"        "idiosyncraticvol"
 [7] "an_booktomarket"    "an_assetgrowth"     "an_cbprofitability"
[10] "ln_turn"
```

4. Fit-the-Fit:
Where are the nonlinearities and interactions ??

In this section we will use abart10 which is $\hat{R} = \hat{f}^A$ using the 10 selected x variables and BART.

In order to understand the fit, we fit trees to the fit.

Could use the out of sample predictions.
Could roll the procedure, could ...

To understand the nonlinearities:
we try pulling out the linear fit from $\hat{R}$ and fit trees to the residuals.

To understand the interactions:
we try pulling out the GAM fit from $\hat{R}$ and fit trees to the residuals.

Fit a simple tree to fit $= \hat{R}$

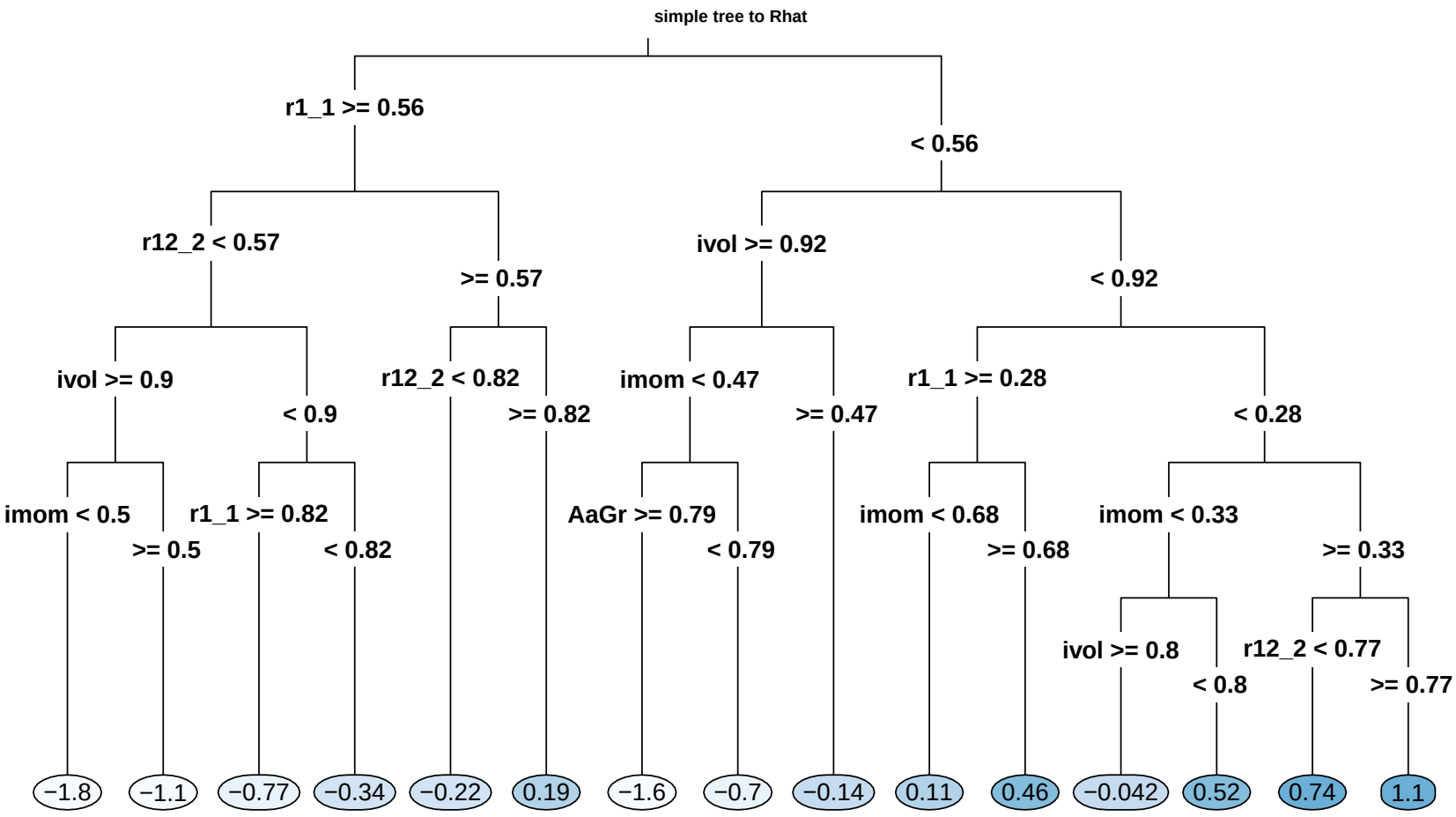
We have no idea what $\hat{R}$ means in terms of the role that the explanatory variables play!!

We first fit a simple tree *to the fit* $\hat{R}$.

Note: There are a lot of trees in $\hat{R} = \hat{f}^A$:
(number of trees in each ensemble) x (number of posterior draws) x (number of months) $=$

```
In [2]: 200*10000*629
Out[2]: 1258000000
```

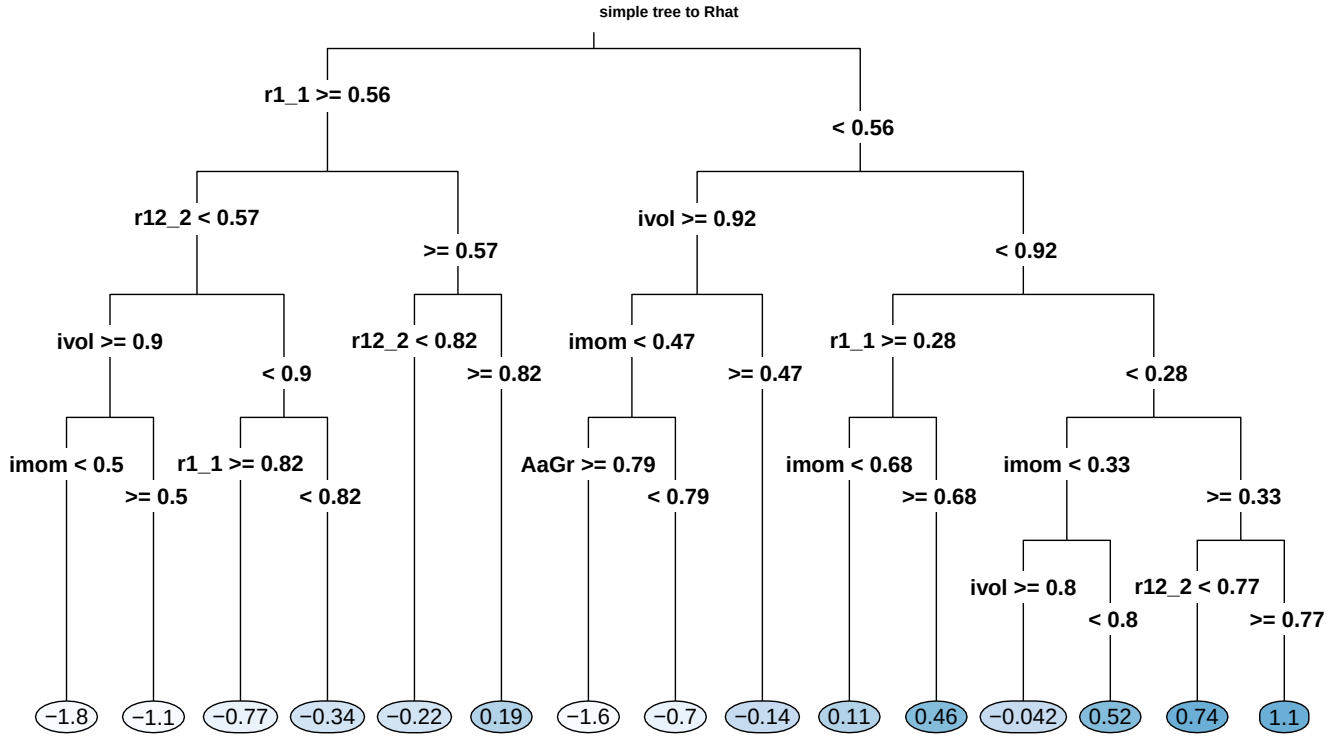
Fit a simple tree to fit $= \hat{R}$



Now we have some idea about the relationship between $\hat{R}$ and x!!

variables used: r1_1, r12_2, ivol, imom, AaGr.

Of course, we may have oversimplified.



To get a low return you need
(going down the left part of the tree):

- `r1_1` big.
- `r12_2` small.
- `ivol` big.
- `imom` small.

To get a high return you need
(going down the right part of the tree):

- `r1_1` small.
- `r12_2` big.
- `ivol` not too big.
- `imom` not too small.

But there are some tricky parts to the tree, nonlinearities, interactions ....

- sort bottom nodes by mean fit.
- display the distribution of each x (row) at each mean fit for a bottom node (column).

Looking for Non-linearities: Fit-the-Fit, Linear residuals

Looking long and hard at the trees can give you a sense of the relationship, but figuring out what is linear and not, is hard.

Our idea is that *mostly* the fit $\hat{R}$ is well approximated by a linear fit.

But, there are important "dusty" corners where there are departures from linearity.

To find the dusty corners, we regress the fit $\hat{R}$ on x and then seek to understand the residuals.

Figuring out the tree relating the fit to x can be hard.

But this is ridiculous.

(the coefficients for the linear fit of the fit).

```
Coefficients:
(Intercept)           me          r1_1         r12_2          imom          seas          ivol
  -0.004287    -0.006047    -0.015658     0.008134     0.007388     0.005134    -0.007636
        btm          AaGr       AcbProf       ln_turn
   0.007095    -0.003096     0.008067     0.006008
```

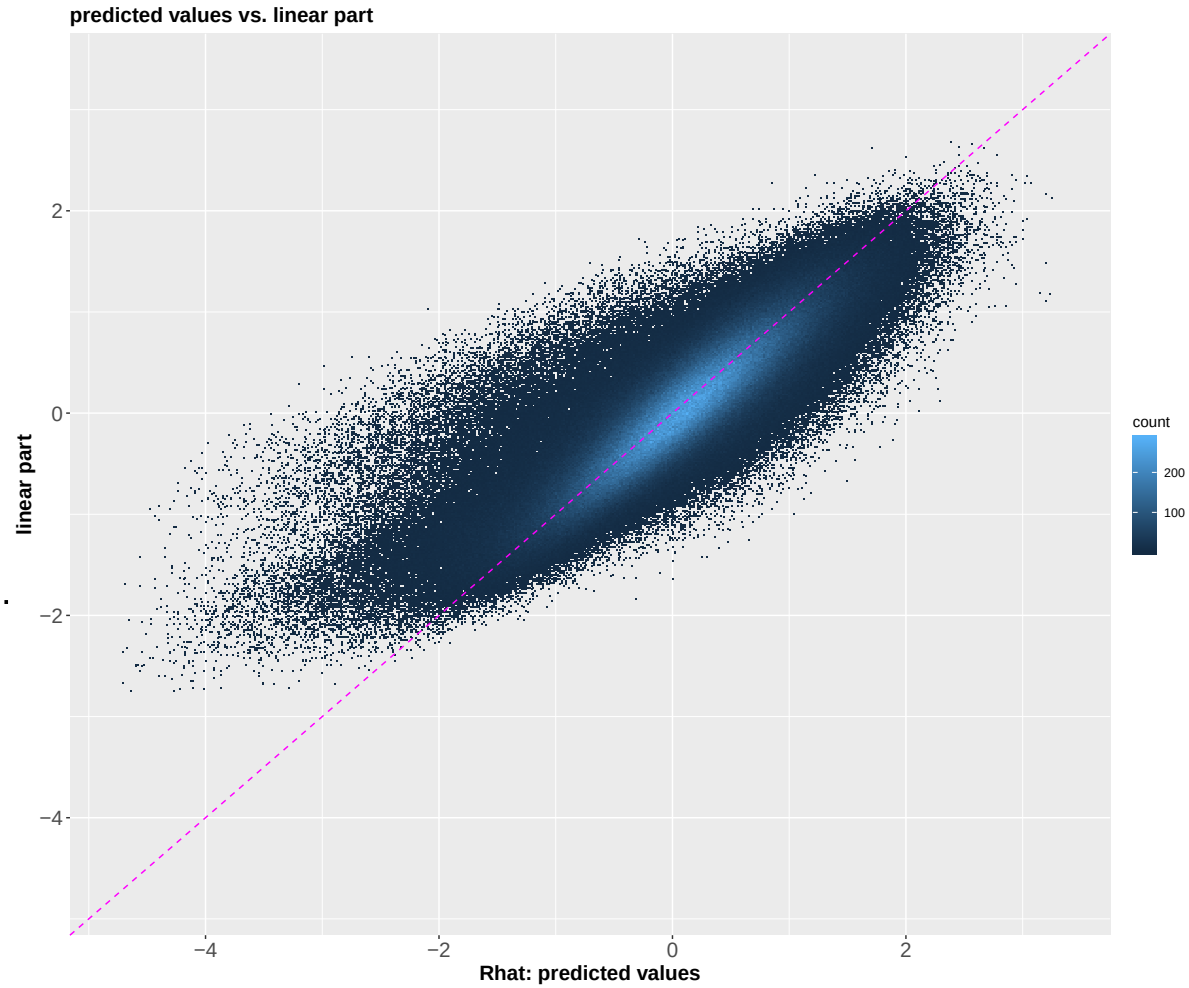
Get the linear and nonlinear parts of the fit $= \hat{R}$

x axis: fit $= \hat{R}$.

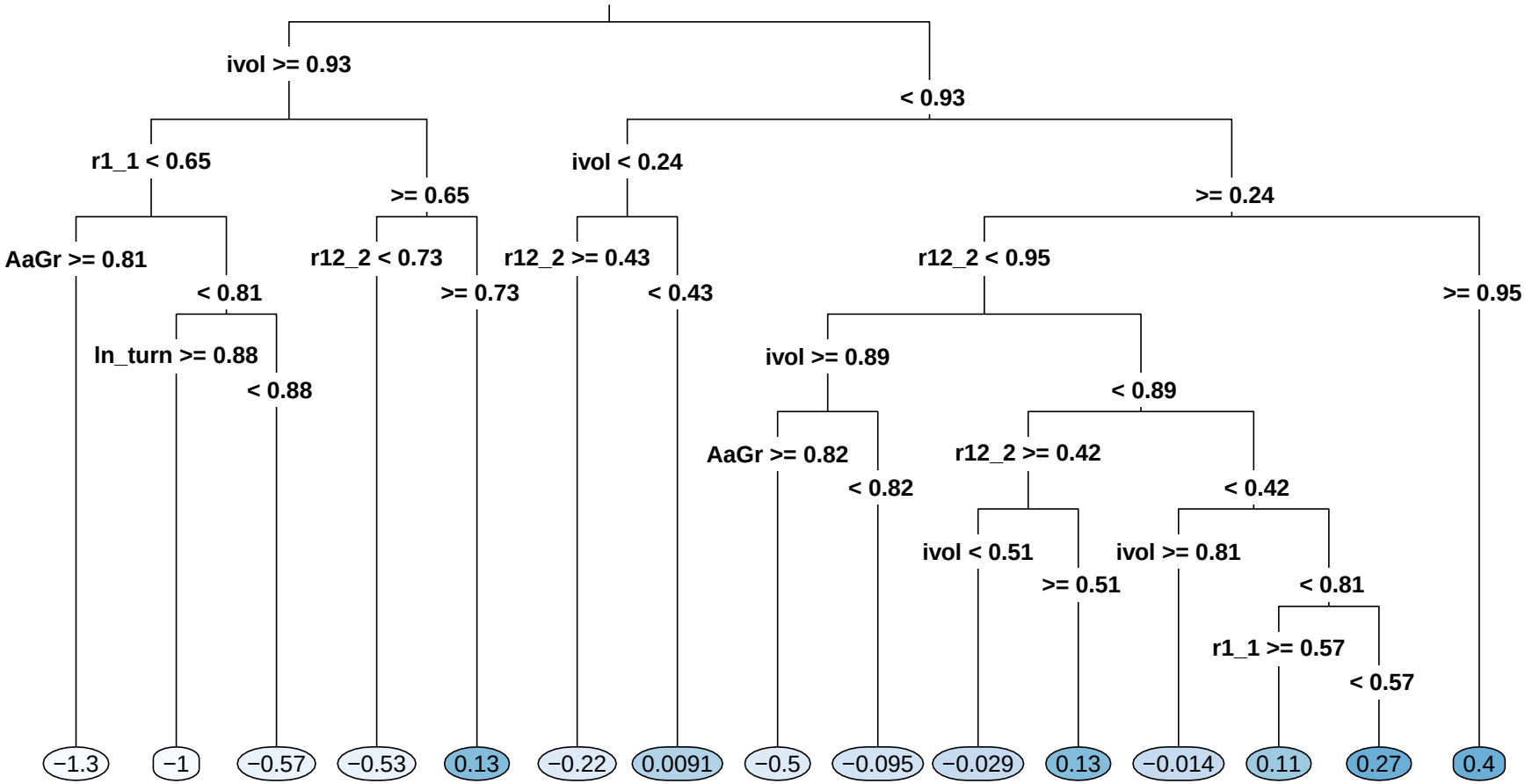
y axis:
fit from linear regression of $\hat{R}$ on x.

We call the residuals
"the nonlinear part of the fit".

Note the asymmetry:
Linear misses the low
more than the high.



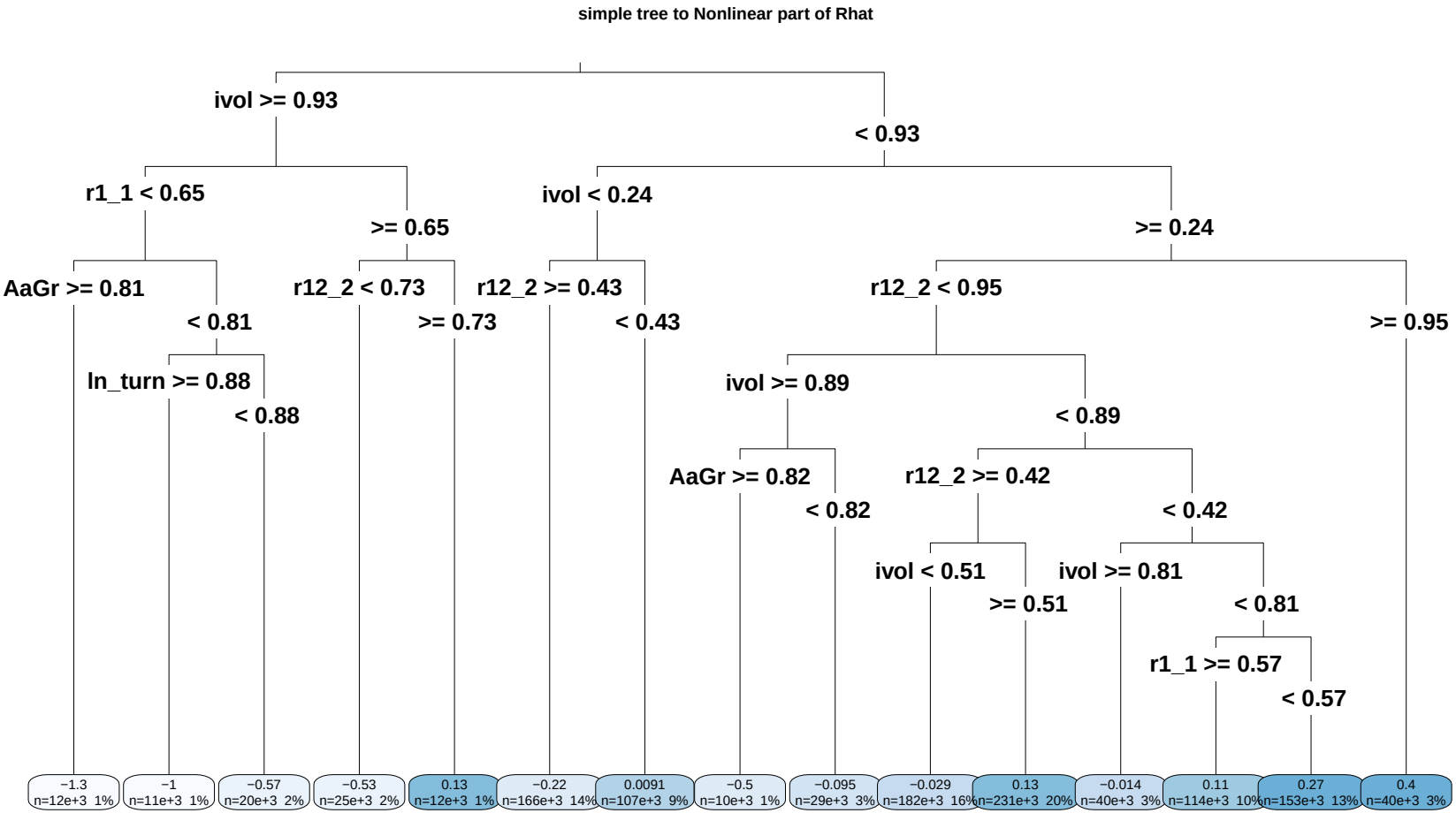
Simple tree fit to the nonlinear part of $\hat{R}$



Now `ivol` is killer, and `ln_turn` comes in.

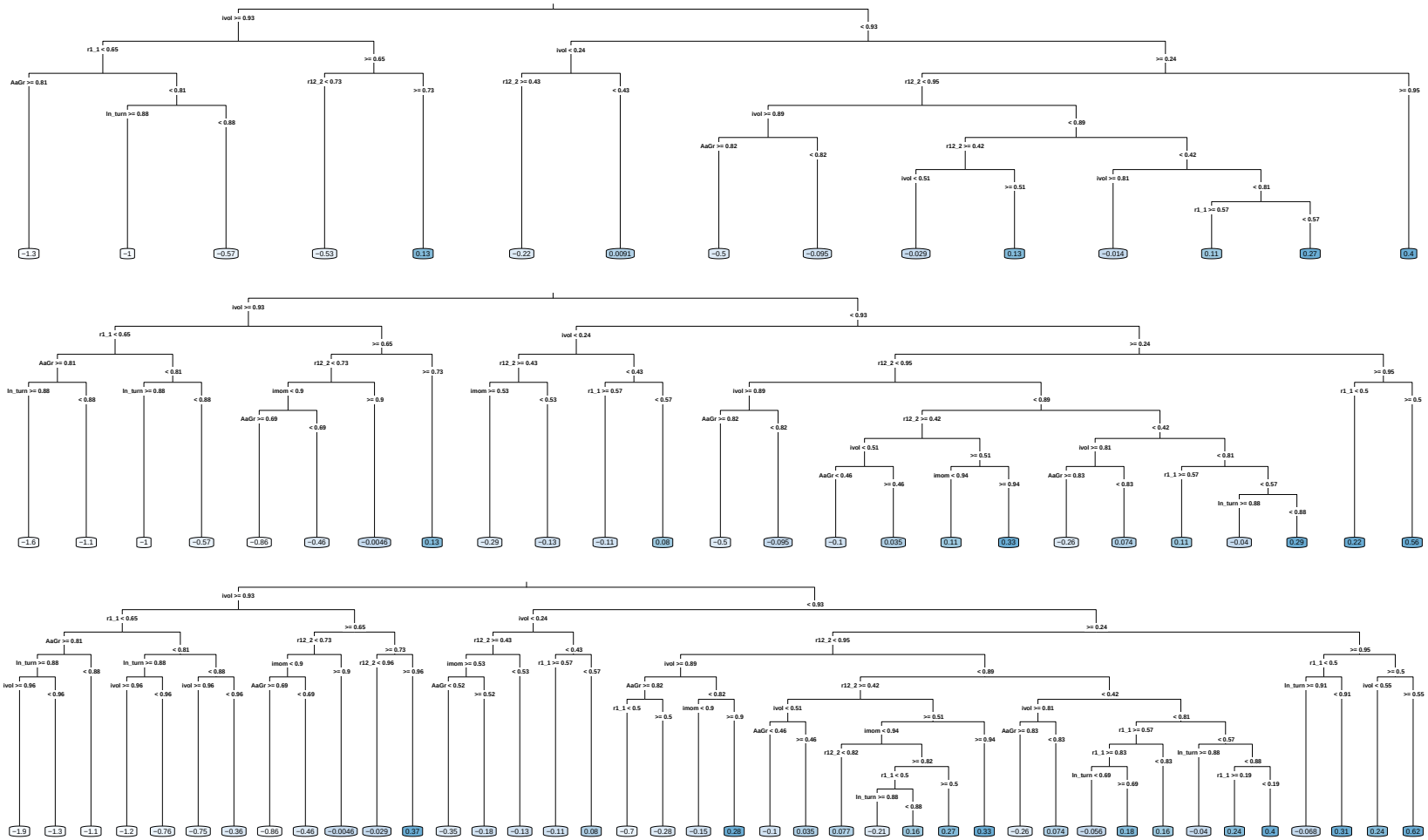
How dusty are the corners??

Here we include the number of observations (and percent) in each bottom node.



trees of various sizes fit to nonlinear part of $\hat{R}$

Trees of size 15, 25, 40.



Looking for Interactions: Fit-the-Fit, GAM Residuals

We have found the parts of predictor space where the nonlinear fit seems to be different from the linear fit.

But *how* are they different??

Something we often think about are *interactions*.

Do certain variables *combine* to produce an effect.

We will pull out a GAM fit and look at the resids to find the interactions.

What is a GAM?

$$f(x_1, x_2, \ldots, x_p) = \sum_{j=1}^{p} f_j(x_j).$$

where we are very flexible in the fitting of each f_j.

So we can be as nonlinear as we like in each variable, but there are no interactions.

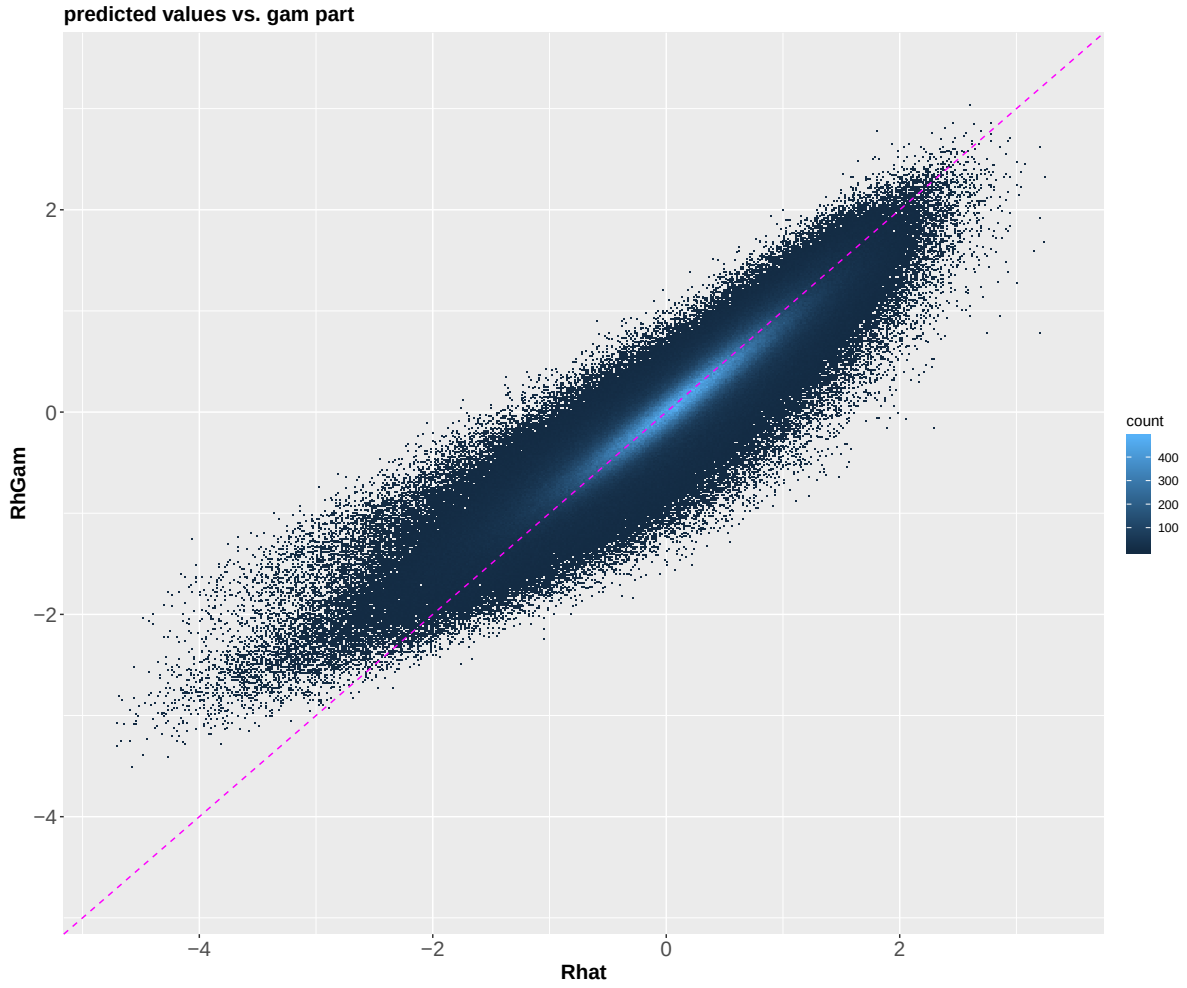
Rhat: $\hat{R}$ using abart10.
RhLin: fits from regression of $\hat{R}$ on x.
RhNLin: residuals from regression of $\hat{R}$ on x.
RhGam: fits from GAM fit of $\hat{R}$ on x.
RhNGam: residuals from GAM fit of $\hat{R}$ on x.

```
> print(round(cor(dfM),digits=3))
        Rhat RhLin RhNLin RhGam RhNGam
Rhat   1.000 0.857  0.516 0.933  0.525
RhLin  0.857 1.000  0.000 0.912  0.184
RhNLin 0.516 0.000  1.000 0.294  0.714
RhGam  0.933 0.912  0.294 1.000  0.185
RhNGam 0.525 0.184  0.714 0.185  1.000
```

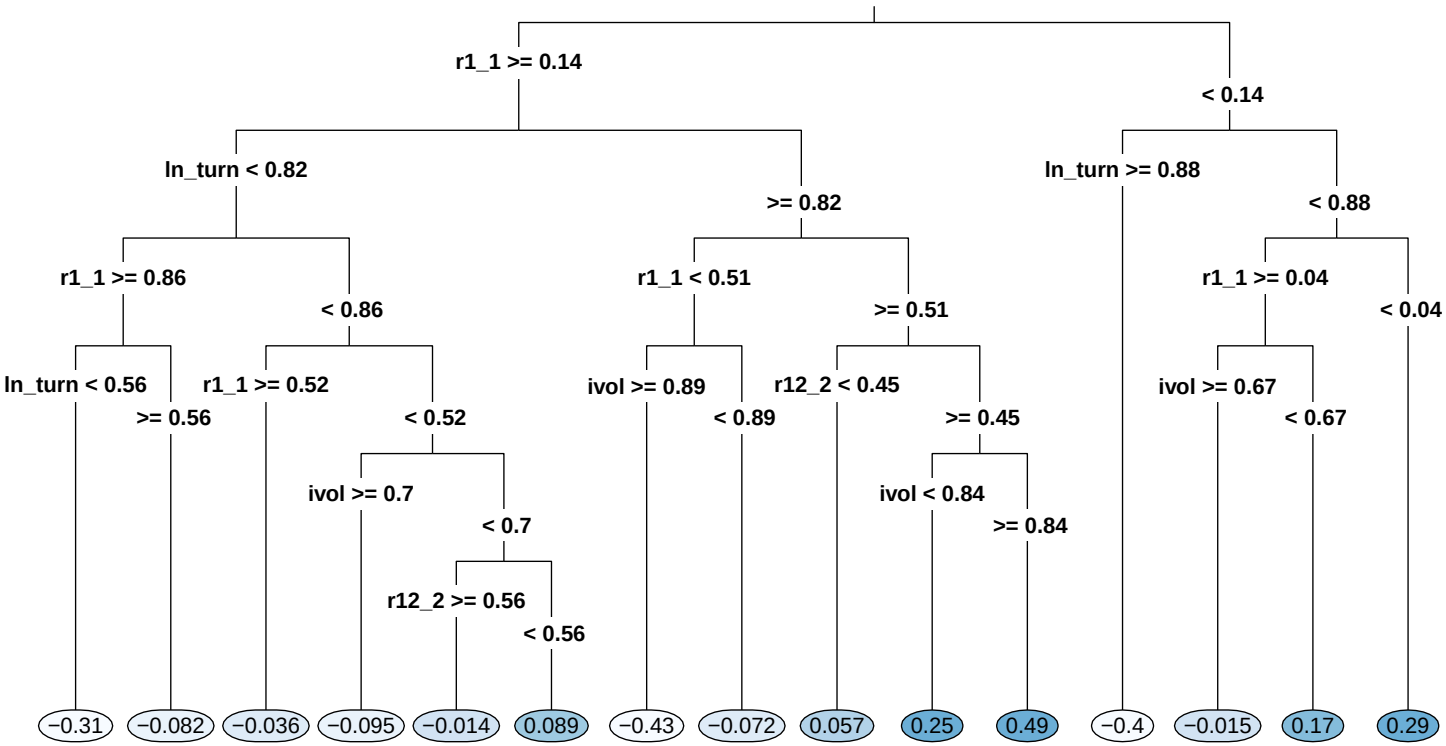
GAM fit of $\hat{R}$

Much better fit than linear.

Still asymmetric, but not as much.

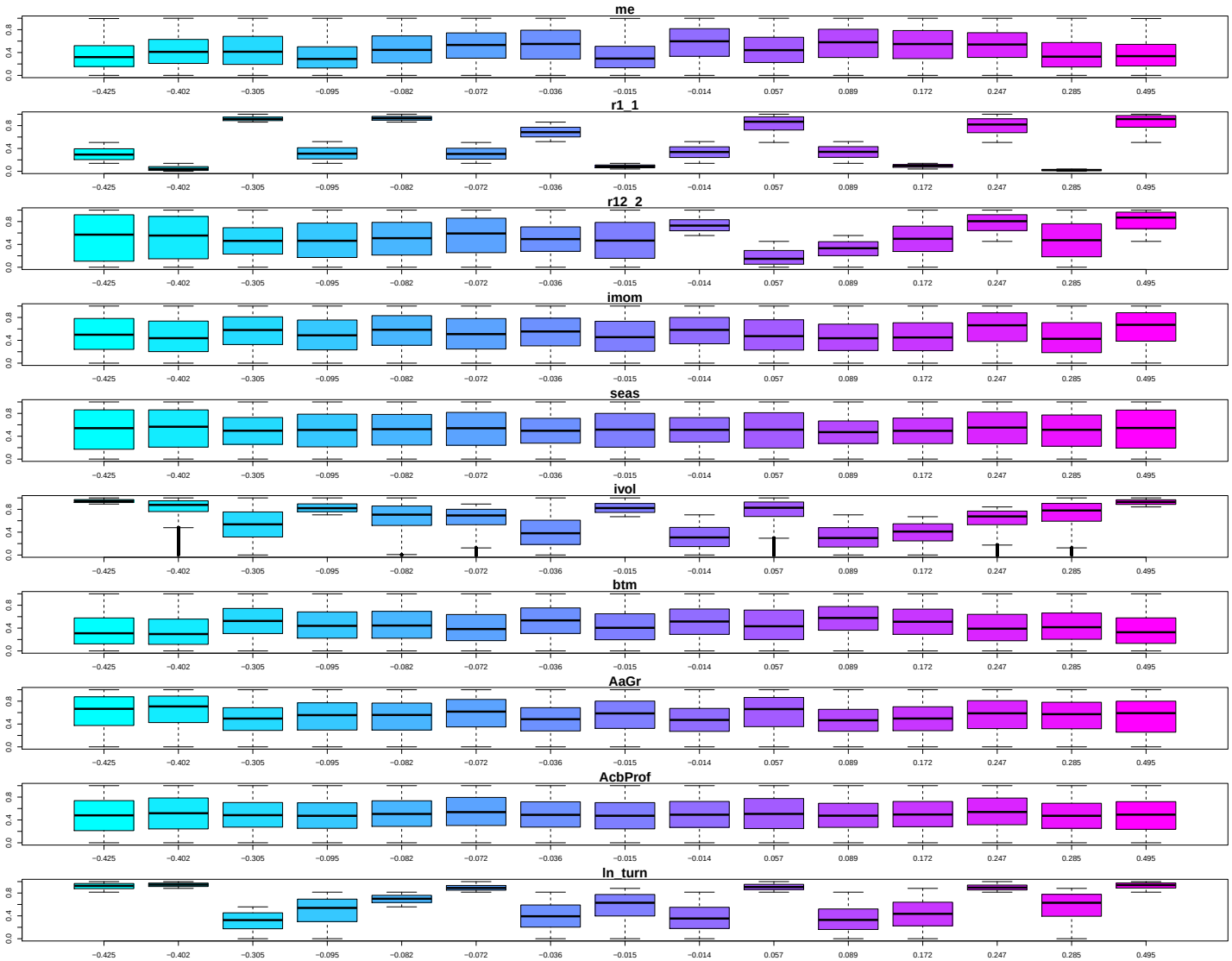


Tree with 15 bottom nodes: fit the resids from GAM fit to $\hat{R}$.



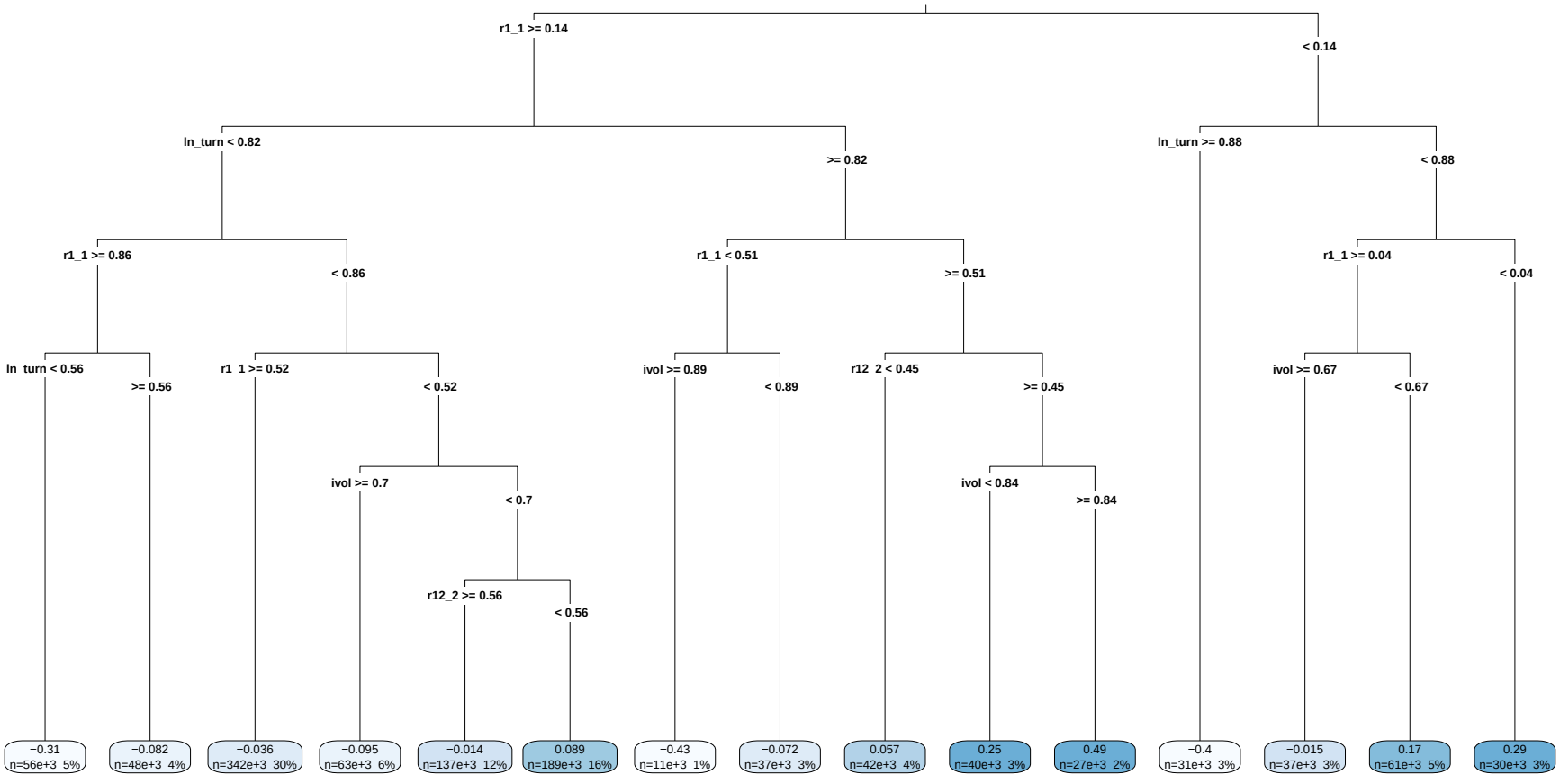
- `ln_turn` and `ivol` are huge.
- interesting tree, look where -.43 and .49 are!
  they both have `ln_turn` $\geq$ .82 and big `ivol` !!

r1_1, ivol, ln_turn, and r12_2 are wild !!!



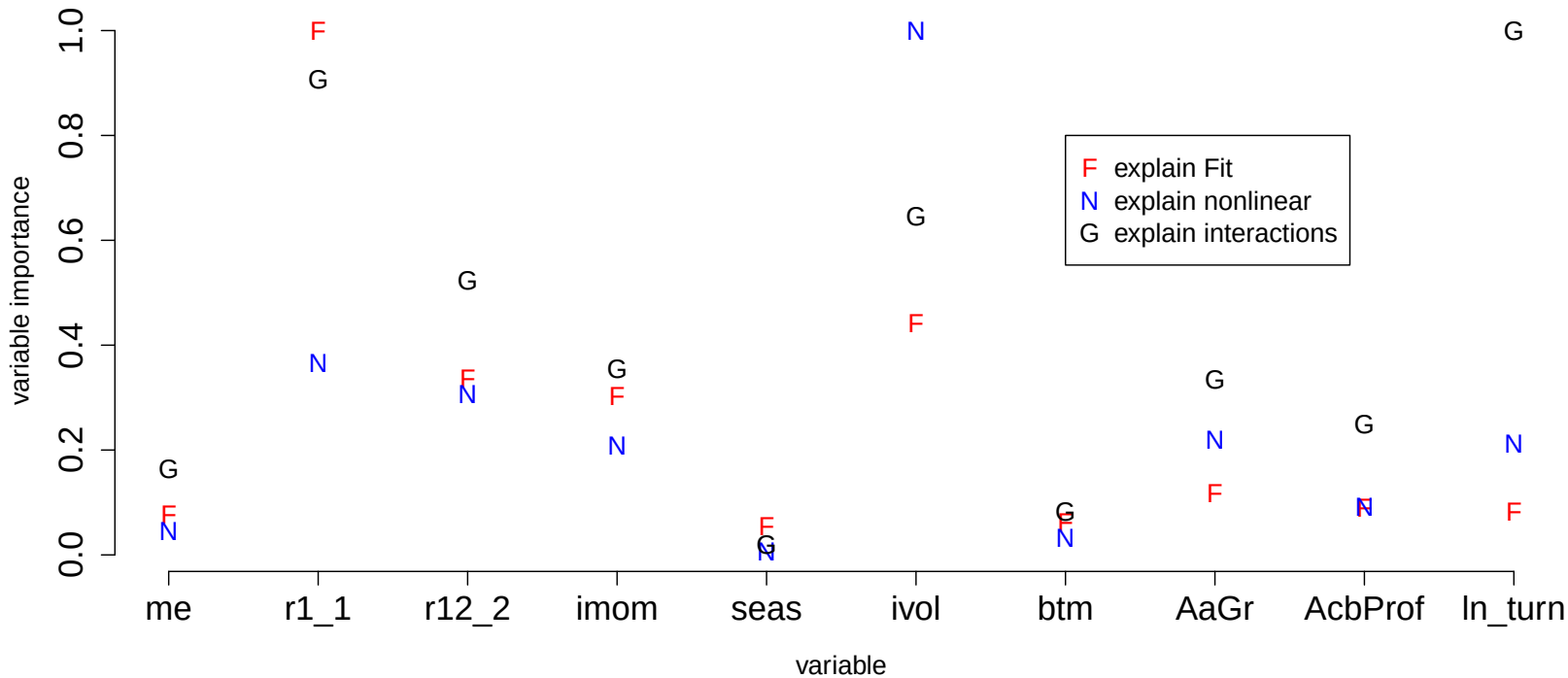
How dusty are the corners ??!!

Each bottom node indicates the number of observations.



Use rpart Variable Importance

- for the fit $\hat{R}$, resids from linear, and resids from GAM.



5. Concluding Remarks

We have focused on a simple Machine Learning approach to get a feeling for the nonlinear relationship between excess returns and predictors.

In a "fairly" simple way we can see things like `r1_1`, `ivol` and `ln_turn` are hugely influential, nonlinear, particular in the dusty corners.