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

Carlos Carvalho, John Cochrane, Juhani Linnainmaa, Rob McCulloch

1. Goals
2. Predictability
3. Variable Selection
4. Fit-the-Fit, Where are the nonlinearities and interactions ??
5. How often they are in the same tree?
6. Sliced Inverse Regression
7. Notes on the Data
8. Concluding Remarks

1. Goals

Predict

Simple approach to predicting the monthly cross section of firm returns using variables obtained in the previous month.

Use ensembles of trees.

Interpret

fit the fit

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

E.g. Variable selection: Can I find a function of a subset of the variables the 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

```
> dim(TrxI)
[1] 1153117      33
> colnames(TrxI)
 [1] "me"                          "r1_1"
 [3] "r12_2"                       "r12_7"
 [5] "industrymom"                 "r60_13"
...
[31] "ln_cvvol"                    "ln_turn"
```

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.

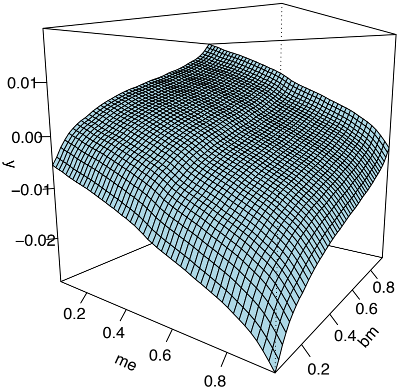
We ran default BART and default random forests.

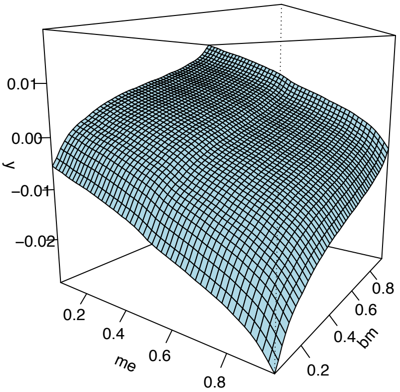
Our goal is to have some understanding of what the non-linear fitted relationship is.

With a two-dimensional x, we can plot.

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

Hard in high dimensions!!!





- Looks pretty linear for most of the middle.
- big me and small bm really interact to give you low returns.
- A little non-linear upturn for big bm, especially at small me.
  At big me, small bm , there is a *dusty corner*.

Note:

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 have some interesting results with neural nets.

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.

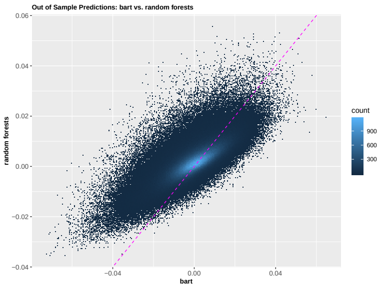
rf is Random Forests.
abart uses the average $\hat{f}$ from all months.
*10 uses just 10 variables we got from our variable selection.
tree used 25 bottom nodes.

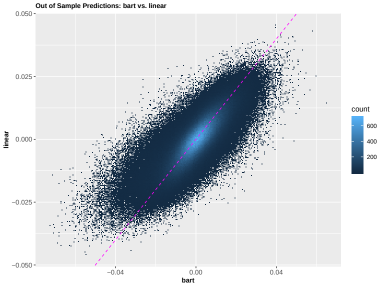
	R	linear	tree	rf	bart	bart10	abart	abart10
R	1.0000	0.0482	0.0409	0.0468	0.0553	0.0572	0.0706	0.0693
linear	0.0482	1.0000	0.5929	0.7160	0.7993	0.7589	0.7850	0.7536
tree	0.0409	0.5929	1.0000	0.7288	0.6414	0.6278	0.5613	0.5565
rf	0.0468	0.7160	0.7288	1.0000	0.7611	0.7147	0.6580	0.6380
bart	0.0553	0.7993	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 predictions compared to linear and Random Forests:

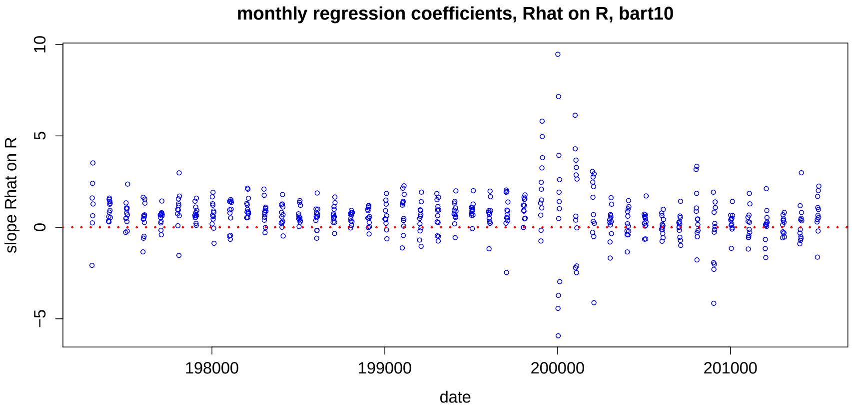
BART is much more like linear.

Different everywhere,
but most different at small returns.



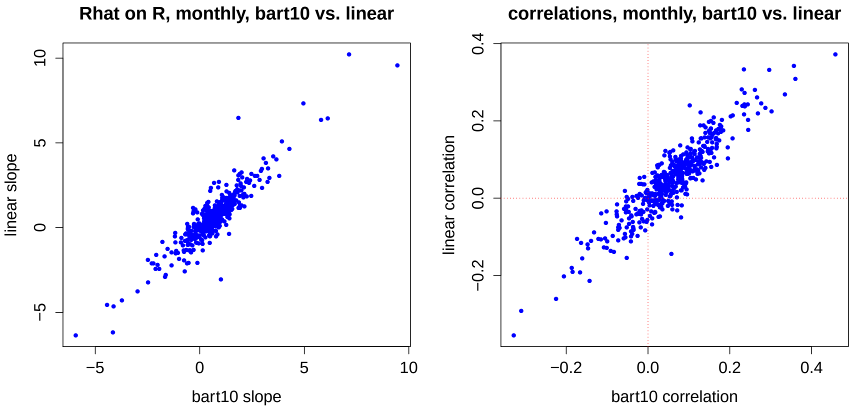


Regress out-of-sample $\hat{R}$ on R each month



What happened in 2000 ????

Compare bart10 and linear, slopes and correlations, each month



- some predictability
- big picture, bart10 like linear

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.

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):

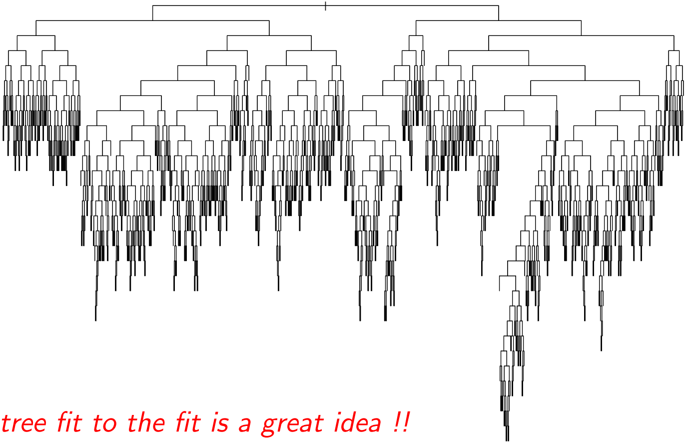
Use backwards and forwards selection to search for subsets.
As in the linear case, can do all subsets for moderate p.

(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.

(2) is the one simple useful idea in the work.

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 !!!!*

So, for example, the first step in forwards is to fit a big tree to each data set:

$$(y = \hat{f}(X^f), X = x_j^f), \;\; j = 1, 2, \ldots, p$$

and then pick the x_j that gives you the best fit.

Note:

You would not want to fit BART at each x_j^f, it is not engineered to fit perfectly.

You would not want to fit an deep neural net at each x_j^f.

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:

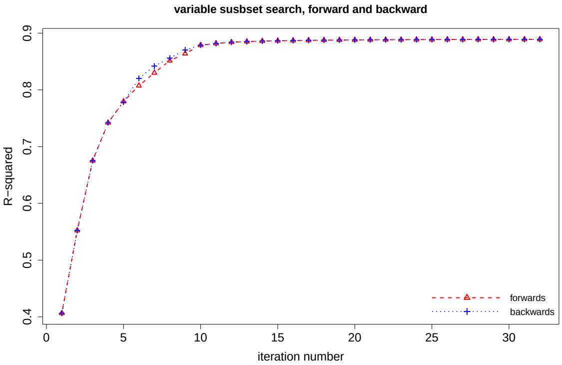
Do the variable selection for each month.

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

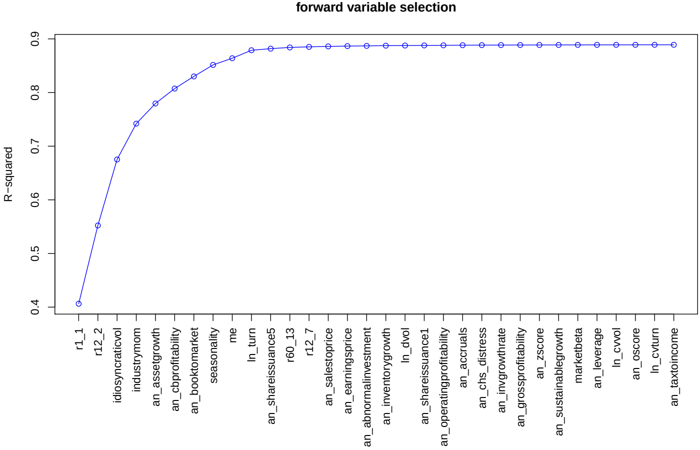
I. Results using $\hat{f}^A$

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.



Gu, Kelly, Xiu:

```
"The most successful predictors are
   price trends, liquidity, and volatility."
```

We agree on those and add a few more.

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"
```

II. Rolled Variable Selection

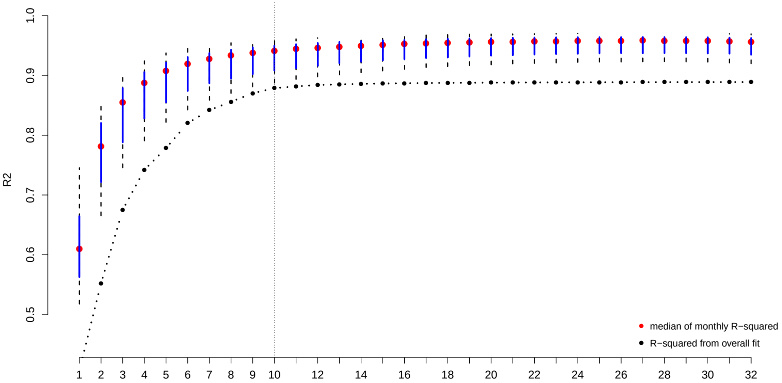
Now we present results for the rolled variable selection.

For each month we seek a nonlinear function of a subset of the variables the approximates the predictions for that month.

The value reported on the y-axis is R^2:

$$corr(\hat{f}(X), \gamma_S(X))^2.$$

The x-axis is $|S|$, the number of variables used.



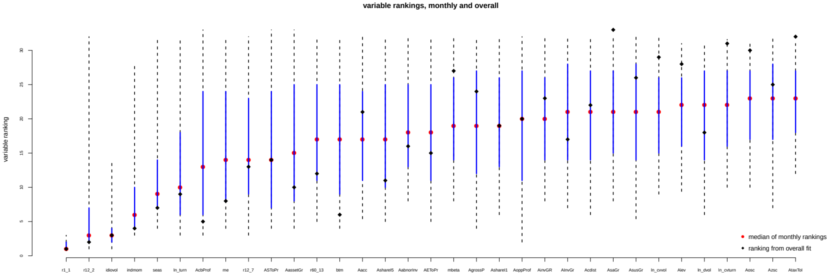
Black dots (the overall $\hat{f}$) suggest that the correlations don't get close to 1, but I tuned it to run fast.
Red is the median R^2 over all the months, blue is 50% of months, black is 90% of months.

We'll use the *rank* of a variable to summarize its importance.

We are using a version of CHM that is like backwards greedy search so rank 1 means the variable was the "last man standing".

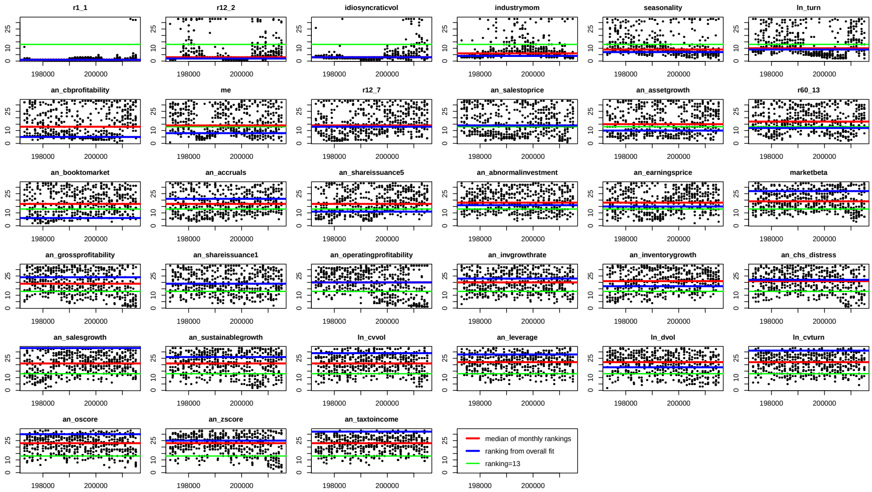
Rank 33 means the variable was the first to be thrown out.

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.



I had to shorten the variable names to fit them in.
Overall and monthly disagree on cash profitability and book-to-market.

Here is the monthly time series of rankings for each variable.



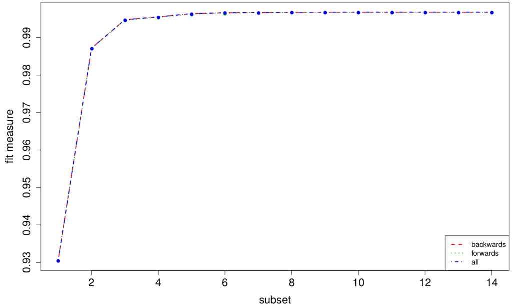
So, it is not completely clear where to draw the line. There is some disagreement on a couple variables (e.g. book to market). However, everyone agrees about the top 13.

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"
```

Note:

Usually, with simple (X, y) data, it looks like this:

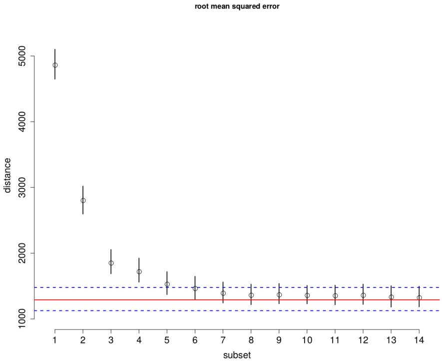


But in this problem we don't get up to .99.
The y=R is very funky, and bart10 did well out of sample.

Note:

In a regular BART fit to (X, y) we can assess the uncertainty using the BART MCMC draws $\{f_d\}$ of the function.

At each subset size we have the posterior distribution of the approximation error.



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.

Note: returns multiplied by 100.

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

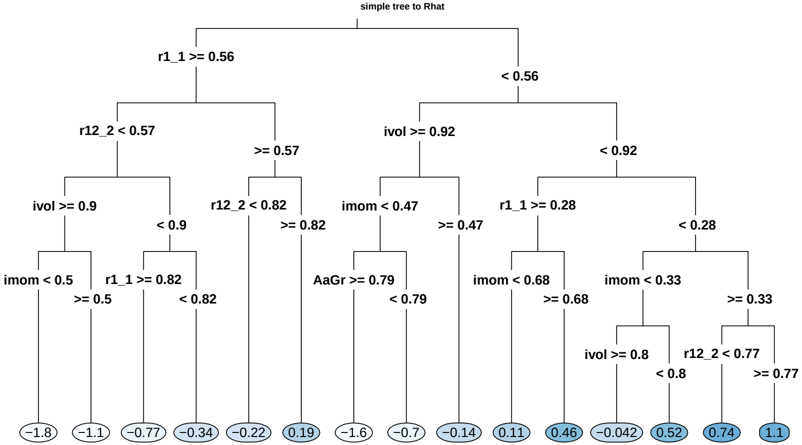
We have no idea what $\hat{R}$ means in terms of the role that the explanatory variables play!!

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
```

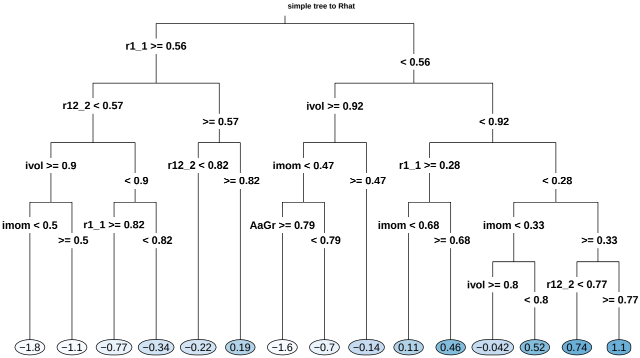
Fit a simple tree to fit $= \hat{R}$



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.

(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
```

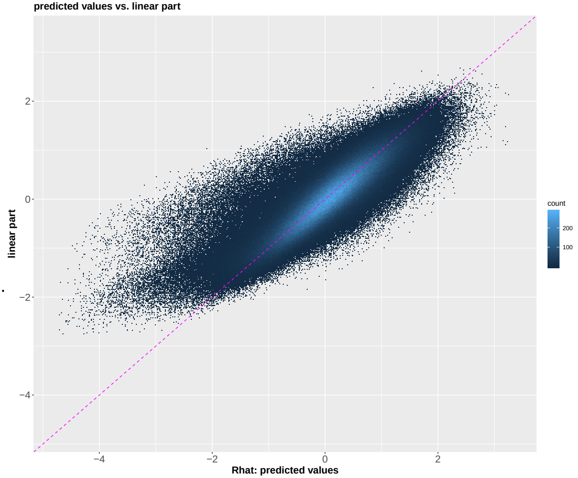
Get the linear and nonlinear parts of the fit $= \hat{R}$

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

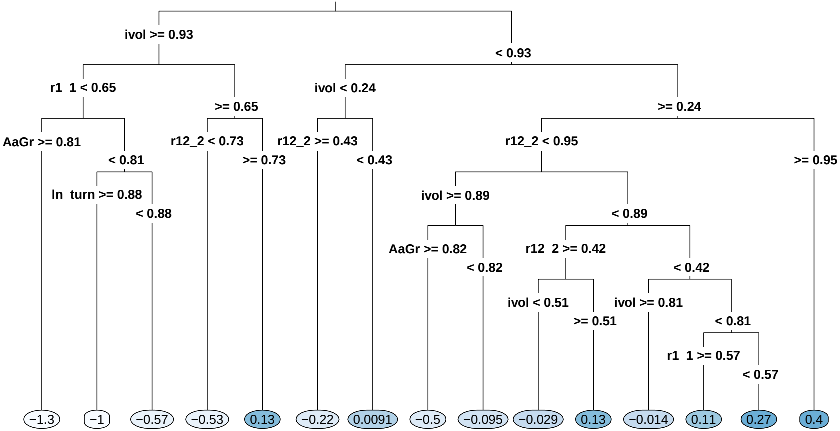
y axis:
fit from linear regression of
$\hat{R}$ on x.

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.

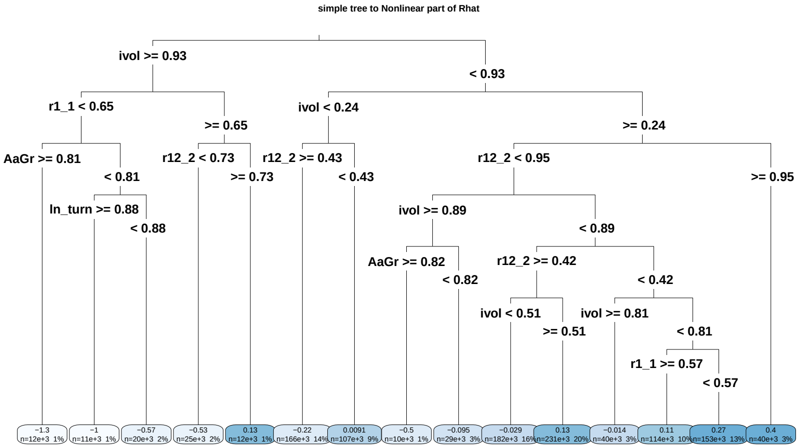
Tree as rules:

Can be easier to understand the tree if we write it out as rules.

```
     R
-1.3486 when ivol >=        0.93                         & r1_1 <  0.65 & AaGr >= 0.81
-1.0290 when ivol >=        0.93                         & r1_1 <  0.65 & AaGr <  0.81 & ln_turn >= 0.88
-0.5720 when ivol >=        0.93                         & r1_1 <  0.65 & AaGr <  0.81 & ln_turn <  0.88
-0.5305 when ivol >=        0.93 & r12_2 <  0.73         & r1_1 >= 0.65
-0.5005 when ivol is 0.89 to 0.93 & r12_2 <  0.95                       & AaGr >= 0.82
-0.2221 when ivol <  0.24        & r12_2 >=         0.43
-0.0952 when ivol is 0.89 to 0.93 & r12_2 <  0.95                       & AaGr <  0.82
-0.0294 when ivol is 0.24 to 0.51 & r12_2 is 0.42 to 0.95
-0.0145 when ivol is 0.81 to 0.89 & r12_2 <  0.42
 0.0091 when ivol <  0.24        & r12_2 <  0.43
 0.1088 when ivol is 0.24 to 0.81 & r12_2 <  0.42         & r1_1 >= 0.57
 0.1302 when ivol is 0.51 to 0.89 & r12_2 is 0.42 to 0.95
 0.1342 when ivol >=        0.93 & r12_2 >=         0.73 & r1_1 >= 0.65
 0.2734 when ivol is 0.24 to 0.81 & r12_2 <  0.42         & r1_1 <  0.57
 0.3961 when ivol is 0.24 to 0.93 & r12_2 >=         0.95
```

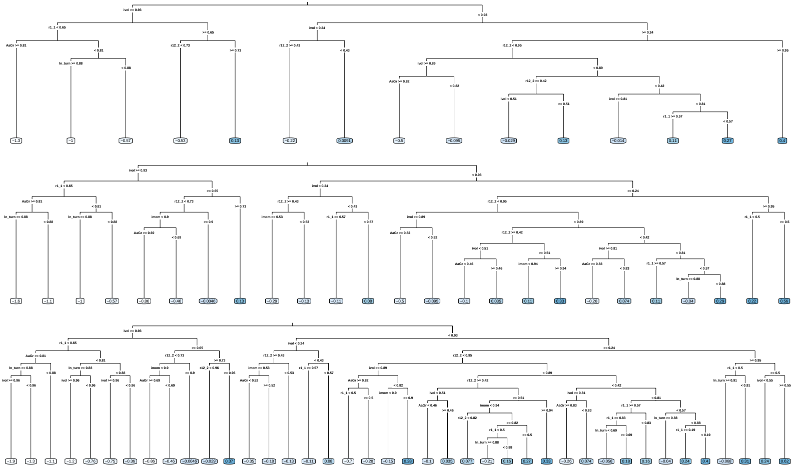
How dusty are the corners??

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.

Pretty popular in applied statistics.

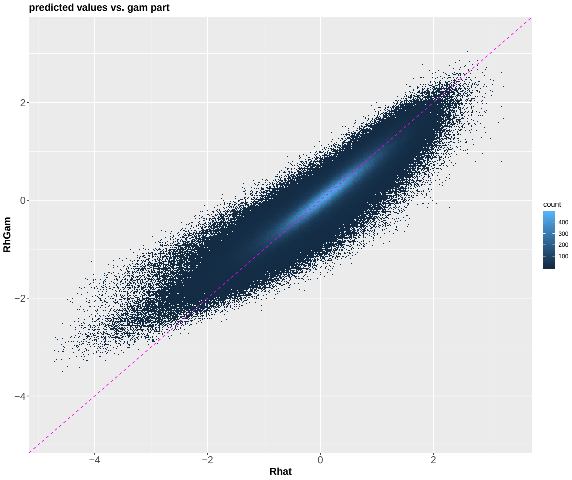
Rhat: $\hat{R}$ using abart10.
RhLin: fits from regression of $\hat{R}$ on x.
RhNLin: residuals from regression of $\hat{R}$ on x.
RhGam: fits from GAM fit of $\hat{R}$ on x.
RhNGam: residuals from GAM fit of $\hat{R}$ on x.

```
> 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
```

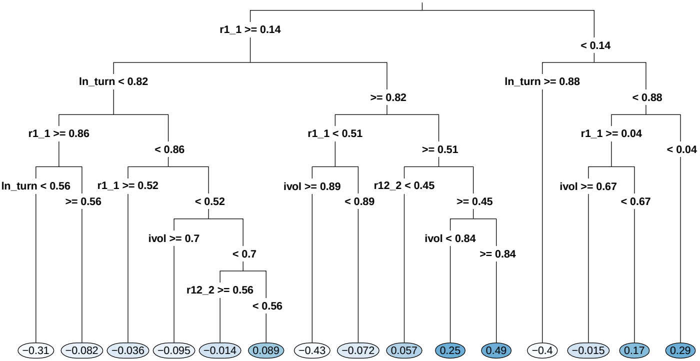
GAM fit of $\hat{R}$

Much better fit than linear.

Still asymmetric, but not as much.



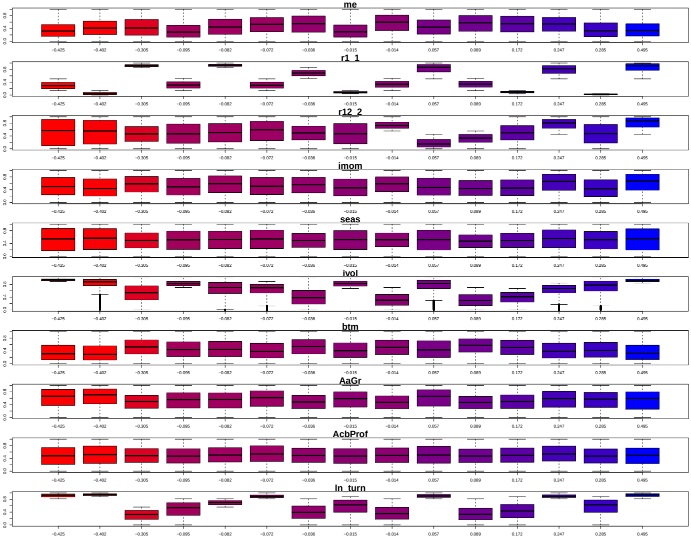
Tree with 15 bottom nodes: fit the resids from GAM fit to $\hat{R}$.



- `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 !!!



Rules for tree with 15 bottom nodes:
fit the resids from GAM fit to $\hat{R}$ from abart10.

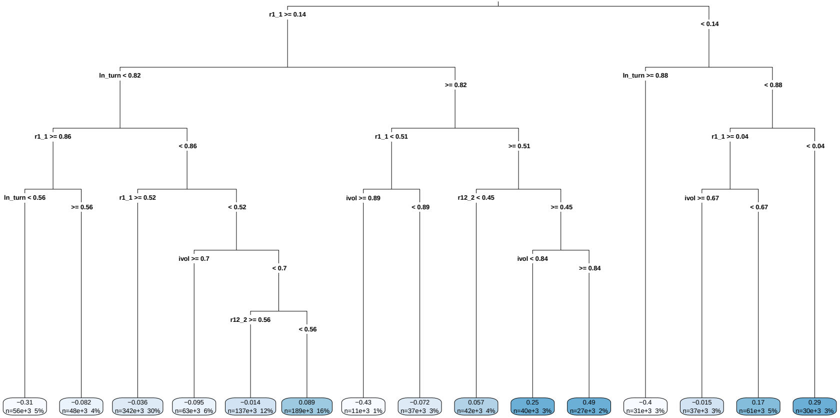
```
RhNGam
 -0.425 when r1_1 is 0.14 to 0.51 & ln_turn >=         0.82 & ivol >= 0.89
 -0.402 when r1_1 <  0.14         & ln_turn >=         0.88
 -0.305 when r1_1 >=         0.86 & ln_turn <  0.56
 -0.095 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol >= 0.70
 -0.082 when r1_1 >=         0.86 & ln_turn is 0.56 to 0.82
 -0.072 when r1_1 is 0.14 to 0.51 & ln_turn >=         0.82 & ivol <  0.89
 -0.036 when r1_1 is 0.52 to 0.86 & ln_turn <  0.82
 -0.015 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol >= 0.67
 -0.014 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol <  0.70 & r12_2 >= 0.56
  0.057 when r1_1 >=         0.51 & ln_turn >=         0.82                & r12_2 <  0.45
  0.089 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol <  0.70 & r12_2 <  0.56
  0.172 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol <  0.67
  0.247 when r1_1 >=         0.51 & ln_turn >=         0.82 & ivol <  0.84 & r12_2 >= 0.45
  0.285 when r1_1 <  0.04         & ln_turn <  0.88
  0.495 when r1_1 >=         0.51 & ln_turn >=         0.82 & ivol >= 0.84 & r12_2 >= 0.45
```

Rules for tree with 25 bottom nodes:
fit the resids from GAM fit to $\hat{R}$ from abart10.

```
RhNGam
-0.845 when r1_1 <  0.14         & ln_turn >=         0.88 & ivol >= 0.81                & AaGr >= 0.66 & imom <  0.5
-0.621 when r1_1 is 0.14 to 0.51 & ln_turn >=         0.82 & ivol >= 0.89                & AaGr >= 0.63
-0.424 when r1_1 <  0.14         & ln_turn >=         0.88 & ivol >= 0.81                & AaGr <  0.66 & imom <  0.5
-0.305 when r1_1 >=         0.86 & ln_turn <  0.56
-0.276 when r1_1 <  0.14         & ln_turn >=         0.88 & ivol <  0.81                               & imom <  0.5
-0.238 when r1_1 <  0.14         & ln_turn >=         0.88                                              & imom >= 0.5
-0.193 when r1_1 >=         0.86 & ln_turn is 0.56 to 0.82              & r12_2 <  0.48
-0.192 when r1_1 is 0.14 to 0.51 & ln_turn >=         0.82 & ivol >= 0.89                & AaGr <  0.63
-0.095 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol >= 0.70
-0.072 when r1_1 is 0.14 to 0.51 & ln_turn >=         0.82 & ivol <  0.89
-0.070 when r1_1 is 0.52 to 0.86 & ln_turn <  0.48
-0.068 when r1_1 >=         0.86 & ln_turn is 0.56 to 0.82 & ivol <  0.84 & r12_2 >= 0.48
-0.047 when r1_1 >=         0.51 & ln_turn >=         0.82              & r12_2 <  0.45 & AaGr >= 0.63
-0.015 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol >= 0.67
-0.014 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol <  0.70 & r12_2 >= 0.56
 0.016 when r1_1 is 0.52 to 0.86 & ln_turn is 0.48 to 0.82
 0.089 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol <  0.70 & r12_2 <  0.56
 0.172 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol <  0.67
 0.173 when r1_1 >=         0.51 & ln_turn >=         0.82              & r12_2 <  0.45 & AaGr <  0.63
 0.247 when r1_1 >=         0.51 & ln_turn >=         0.82 & ivol <  0.84 & r12_2 >= 0.45
 0.264 when r1_1 is 0.51 to 0.79 & ln_turn >=         0.82 & ivol >= 0.84 & r12_2 >= 0.45
 0.266 when r1_1 >=         0.86 & ln_turn is 0.56 to 0.82 & ivol >= 0.84 & r12_2 >= 0.48
 0.285 when r1_1 <  0.04         & ln_turn <  0.88
 0.380 when r1_1 >=         0.79 & ln_turn >=         0.82 & ivol >= 0.84 & r12_2 >= 0.45               & imom <  0.5
 0.684 when r1_1 >=         0.79 & ln_turn >=         0.82 & ivol >= 0.84 & r12_2 >= 0.45               & imom >= 0.5
```

How dusty are the corners ??!!

Each bottom node indicates the number of observations.

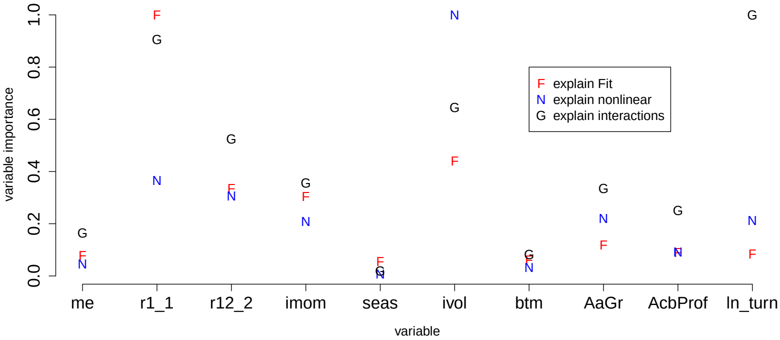


Rules 40 bottom nodes:

```
 RhNGam
-1.0389 when r1_1 <  0.14         & ln_turn >=         0.88 & ivol >=         0.81                         & AaGr >= 0.66 & imom <  0.50 & AcbProf <  0.59
-0.6206 when r1_1 is 0.14 to 0.51 & ln_turn >=         0.82 & ivol >=         0.89                         & AaGr >= 0.63
-0.5655 when r1_1 <  0.14         & ln_turn >=         0.88 & ivol >=         0.81                         & AaGr >= 0.66 & imom <  0.50 & AcbProf >= 0.59
-0.4530 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol >=         0.86                         & AaGr >= 0.59 & imom <  0.65
-0.4235 when r1_1 <  0.14         & ln_turn >=         0.88 & ivol >=         0.81                         & AaGr <  0.66 & imom <  0.50
-0.3966 when r1_1 >=         0.92 & ln_turn <  0.56
-0.2760 when r1_1 <  0.14         & ln_turn >=         0.88 & ivol <  0.81                                                &  imom <  0.50
-0.2415 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol >=         0.67                         & AaGr >= 0.50                & AcbProf <  0.36
-0.2376 when r1_1 <  0.14         & ln_turn >=         0.88                                                              & imom >= 0.50
-0.2273 when r1_1 is 0.86 to 0.92 & ln_turn <  0.56
-0.1930 when r1_1 >=         0.86 & ln_turn is 0.56 to 0.82                        & r12_2 <  0.48
-0.1918 when r1_1 is 0.14 to 0.51 & ln_turn >=         0.82 & ivol >=         0.89                         & AaGr <  0.63
-0.1868 when r1_1 >=         0.51 & ln_turn >=         0.82                        & r12_2 <  0.45         & AaGr >= 0.63                & AcbProf <  0.45
-0.1462 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol >=         0.86                         & AaGr <  0.59 & imom <  0.65
-0.1228 when r1_1 is 0.73 to 0.86 & ln_turn <  0.48
-0.0766 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol is 0.70 to 0.86                                        & imom <  0.65
-0.0719 when r1_1 is 0.14 to 0.51 & ln_turn >=         0.82 & ivol <  0.89
-0.0683 when r1_1 >=         0.86 & ln_turn is 0.56 to 0.82 & ivol <  0.84         & r12_2 >=         0.48
-0.0447 when r1_1 is 0.52 to 0.73 & ln_turn <  0.48
-0.0403 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol <  0.70         & r12_2 >=         0.56 & AaGr <  0.66
-0.0228 when r1_1 is 0.52 to 0.86 & ln_turn is 0.48 to 0.82                        & r12_2 <  0.35                                                         & me <  0.41
-0.0157 when r1_1 is 0.52 to 0.86 & ln_turn is 0.48 to 0.82                                                                                                & me >= 0.41
-0.0100 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol >=         0.67                         & AaGr >= 0.50                & AcbProf >= 0.36
 0.0089 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol >=         0.70                                        & imom >= 0.65
 0.0618 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol <  0.70         & r12_2 >=         0.56 & AaGr >= 0.66
 0.0637 when r1_1 >=         0.51 & ln_turn >=         0.82                        & r12_2 <  0.45         & AaGr >= 0.63                & AcbProf >= 0.45
 0.0719 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol <  0.70         & r12_2 is 0.16 to 0.56
 0.1049 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol >=         0.67                         & AaGr <  0.50
 0.1336 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol <  0.67         & r12_2 >=         0.27
 0.1378 when r1_1 is 0.52 to 0.86 & ln_turn is 0.48 to 0.82                        & r12_2 >=         0.35                                                 & me <  0.41
 0.1610 when r1_1 <  0.04         & ln_turn is 0.66 to 0.88
 0.1733 when r1_1 >=         0.51 & ln_turn >=         0.82                        & r12_2 <  0.45         & AaGr <  0.63
 0.1734 when r1_1 is 0.14 to 0.52 & ln_turn <  0.82         & ivol <  0.70         & r12_2 <  0.16
 0.2472 when r1_1 >=         0.51 & ln_turn >=         0.82 & ivol <  0.84         & r12_2 >=         0.45
 0.2641 when r1_1 is 0.51 to 0.79 & ln_turn >=         0.82 & ivol >=         0.84 & r12_2 >=         0.45
 0.2659 when r1_1 >=         0.86 & ln_turn is 0.56 to 0.82 & ivol >=         0.84 & r12_2 >=         0.48
 0.2889 when r1_1 is 0.04 to 0.14 & ln_turn <  0.88         & ivol <  0.67         & r12_2 <  0.27
 0.3802 when r1_1 >=         0.79 & ln_turn >=         0.82 & ivol >=         0.84 & r12_2 >=         0.45                & imom <  0.50
 0.3882 when r1_1 <  0.04         & ln_turn <  0.66
 0.6840 when r1_1 >=         0.79 & ln_turn >=         0.82 & ivol >=         0.84 & r12_2 >=         0.45                & imom >= 0.50
```

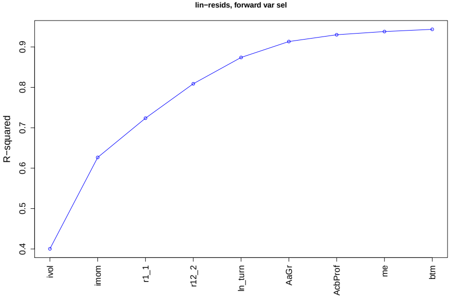
Use rpart Variable Importance

- for the fit $\hat{R}$, resids from linear, and resids from GAM.
- fit a tree of size 1,000 using `rpart`.
- use the variable importance from `rpart`.
- divide each importance by the max (over variables) so numbers are in (0,1].

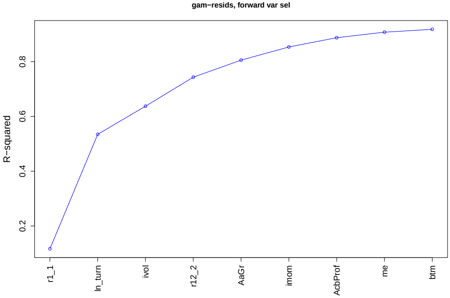


CHM variable selection

top:
resids from linear.



bot:
resids from GAM.



5. How often they are in the same tree?

The BART model is like boosting in that the function is represented as a sum of binary trees.

To get an interaction between two variables, you need them in the same tree.

Unlike boosting, BART gives you the full Bayesian posterior of the tree ensemble.

We sort pairs of variables by how often they are in the same tree.

```
      [,1]                  [,2]
 [1,] "idiosyncraticvol"   "industrymom"
 [2,] "idiosyncraticvol"   "r1_1"
 [3,] "idiosyncraticvol"   "r12_2"
 [4,] "idiosyncraticvol"   "seasonality"
 [5,] "an_assetgrowth"     "idiosyncraticvol"
 [6,] "an_booktomarket"    "idiosyncraticvol"
 [7,] "ln_turn"            "idiosyncraticvol"
 [8,] "industrymom"        "r12_2"
 [9,] "r12_2"              "r1_1"
[10,] "an_cbprofitability" "idiosyncraticvol"
[11,] "idiosyncraticvol"   "me"
[12,] "ln_turn"            "r12_2"
[13,] "seasonality"        "r12_2"
[14,] "ln_turn"            "industrymom"
...
...
[42,] "an_assetgrowth"     "me"
[43,] "an_assetgrowth"     "an_booktomarket"
[44,] "an_cbprofitability" "me"
[45,] "an_cbprofitability" "an_assetgrowth"
```

This says `ivol` is killer.

6. Sliced Inverse Regression

We use $\hat{f}^A$ computed at all $n =$1,153,117 x vectors.

We split the data into 200 groups using the quantiles of the $\{\hat{f}^A(x)\}$ values. That is, group 1 contains the stocks with the lowest predicted returns; group 200 contains those with the highest predicted returns.

Within each group we look at the distribution of the predictor variables, in particular we compute the median of each x.

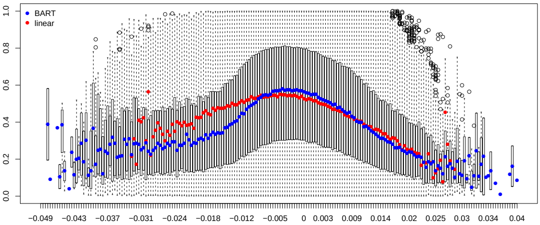
We graphically display how the predictor variables vary over the quantiles of $E(R) \approx \hat{f}^A$.

We use this “inverse-regression” methodology to ask what kind of x do we get given $\hat{R}$.

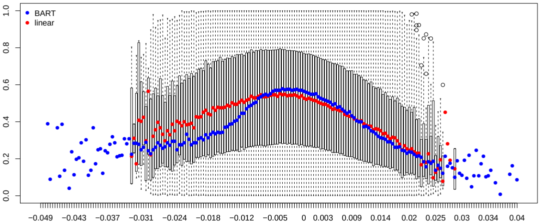
Note that these are not “conditional plots” that report the change in expected R due to a change in one variable with the others help fixed. All the variables move jointly given changes in $E(R)$.

Just me:

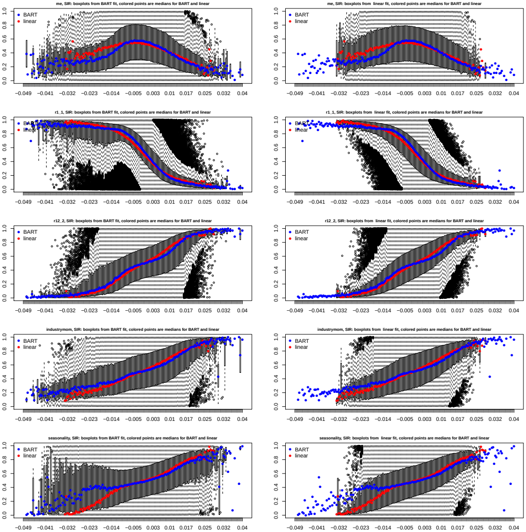
SIR: boxplots from BART fit, colored points are medians for BART and linear



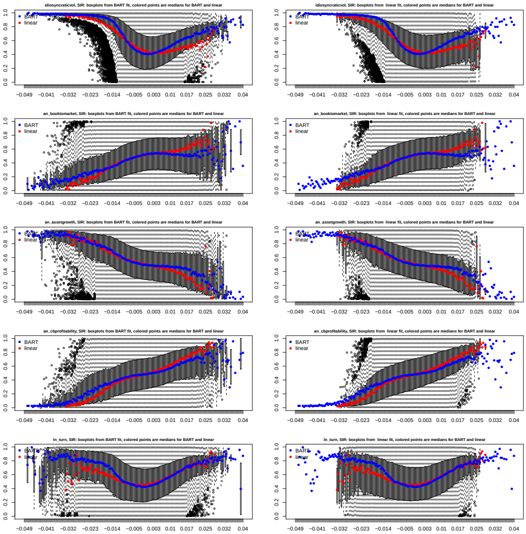
SIR: boxplots from linear fit, colored points are medians for BART and linear



Top 5 variables:



Next 5 variables:



7. Notes on the Data

Start with 3,018,077 observations.
33 “x” variables.

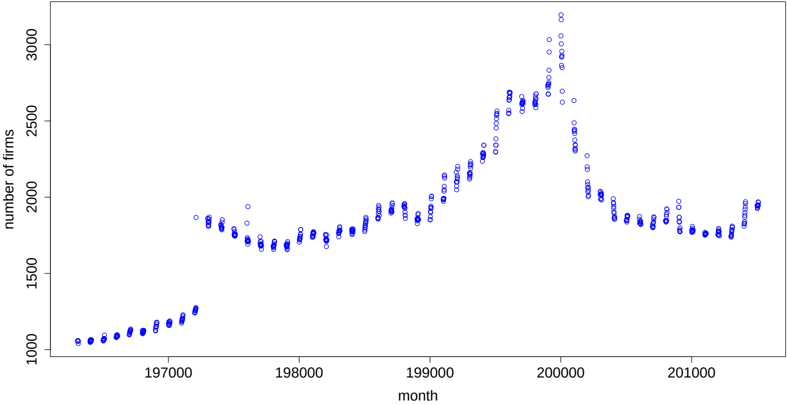
Months:
196306 - 201512.
Throw out “tiny” firms, throw out missing on y=return:

leaves:
n = 1,193,625.
630 months.

Threw out 199509, too many missing values:
629 months.

```
> print(dim(anomd))
[1] 3018077      37
> print(names(anomd))
 [1] "yyyymm"                     "permno"
 [3] "size_cat"                   "retnm"
 [5] "me"                         "r1_1"
 [7] "r12_2"                      "r12_7"
 [9] "industrymom"                "r60_13"
[11] "seasonality"                "marketbeta"
[13] "idiosyncraticvol"           "an_booktomarket"
[15] "an_accruals"                "an_assetgrowth"
[17] "an_abnormalinvestment"      "an_grossprofitability"
[19] "an_operatingprofitability"  "an_cbprofitability"
[21] "an_earningsprice"           "an_salestoprice"
[23] "an_inventorygrowth"         "an_leverage"
[25] "an_oscore"                  "an_zscore"
[27] "an_chs_distress"            "an_salesgrowth"
[29] "an_shareissuance1"          "an_shareissuance5"
[31] "an_sustainablegrowth"       "an_taxtoincome"
[33] "an_invgrowthrate"           "ln_dvol"
[35] "ln_cvvol"                   "ln_turn"
[37] "ln_cvturn"
```

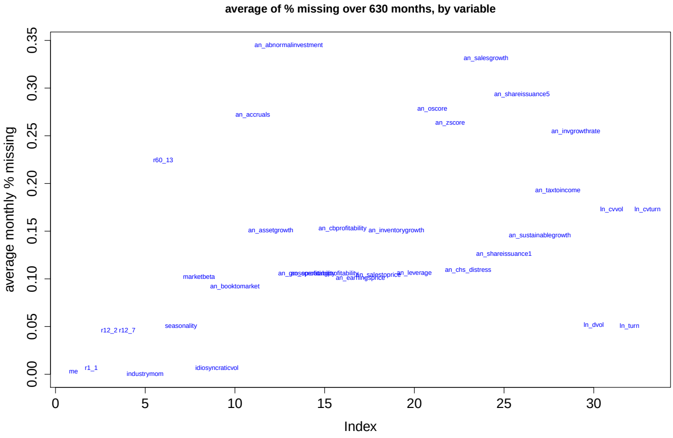
Number of firms in each month:



For each month:

- demean the returns (subtract off the average)
- transform each x using the empirical CDF,
  $\Rightarrow$ each x value $\in [0,1]$.

Missing Values:



Imputed missing values using linear regression.

8. 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 `ivol` and `ln_turn` are hugely nonlinear, particular in the dusty corners.

There is not going to be an easy way to do this!!!

That is why some folks cling to linear.

Could be kidding myself since I may not want trust by fit in the dusty corners!!

Should do nonlinear investigation with more that 10!!

a quote from Gu, Kelly, Xiu:

```
"The most successful predictors are
   price trends, liquidity, and volatility."
```

So, big picture we agree with Gu et. al. but add a few more.

Nice confirmation since much of what we done is different *and* we have much more of a feeling for what kinds or roles the key variables play.

Much more to do:

Will have to try rolling monotone BART, DPMBART, and nnets.

Maybe the basic tree approaches overreact to all the outliers.

Can we assess the uncertainty? DPMBART?