ETC1010: Advanced Modeling

Distributions, Inference, Diagnostics, Decision trees

Di Cook (dicook@monash.edu, @visnut)

3/10/2017

ETC1010: Advanced Modeling

Distributions, Inference, Diagnostics, Decision trees

Di Cook (dicook@monash.edu, @visnut)

3/10/2017

Outline

- Linear model diagnostics
- Regression trees, model by optimisation
- Fit all possible models
- Using linear models for exploration

Linear model diagnostics

- Response variable, y
- Predictors, or explanatory variables, $x_1, \ldots, x_p$
- Assumptions, residual diagnostics
- R^2, deviance, AIC, BIC, likelihood
- Statistically significant, and variable selection

Multiple regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, \ldots, n$$

where ε is a sample from a normal distribution, $N(0, \sigma^2)$.

By optimisation, of $\sum_i (y_i - (b_0 + b_1 x_{i1} + \cdots + b_p x_{ip}))^2$, we found the line of best fit, and parameter estimates $(b_0, b_1, \ldots, b_p, \hat{\sigma})$ for the "true" (population) model.

For a simple linear model,

$$b_1 = r \frac{s_y}{s_y}$$

is the *slope*

$$b_0 = \bar{y} - b_1 \bar{x}$$

is the *intercept*, and

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_{i1} + \cdots + b_p x_{ip}))^2$$

is also called the *mean squared error*.

Population vs sample

Often the data we have is a sample from all possible values available in a larger population. Using the sample we would like to be able to say something about the patterns in the entire population.

Population <->	**Sample**
parameters	statistics
β_k	b_k
σ	s
μ	$\bar{x}$

The population parameters are unknown. The statistics are calculated from the sample, so are known. The model should be written this way:

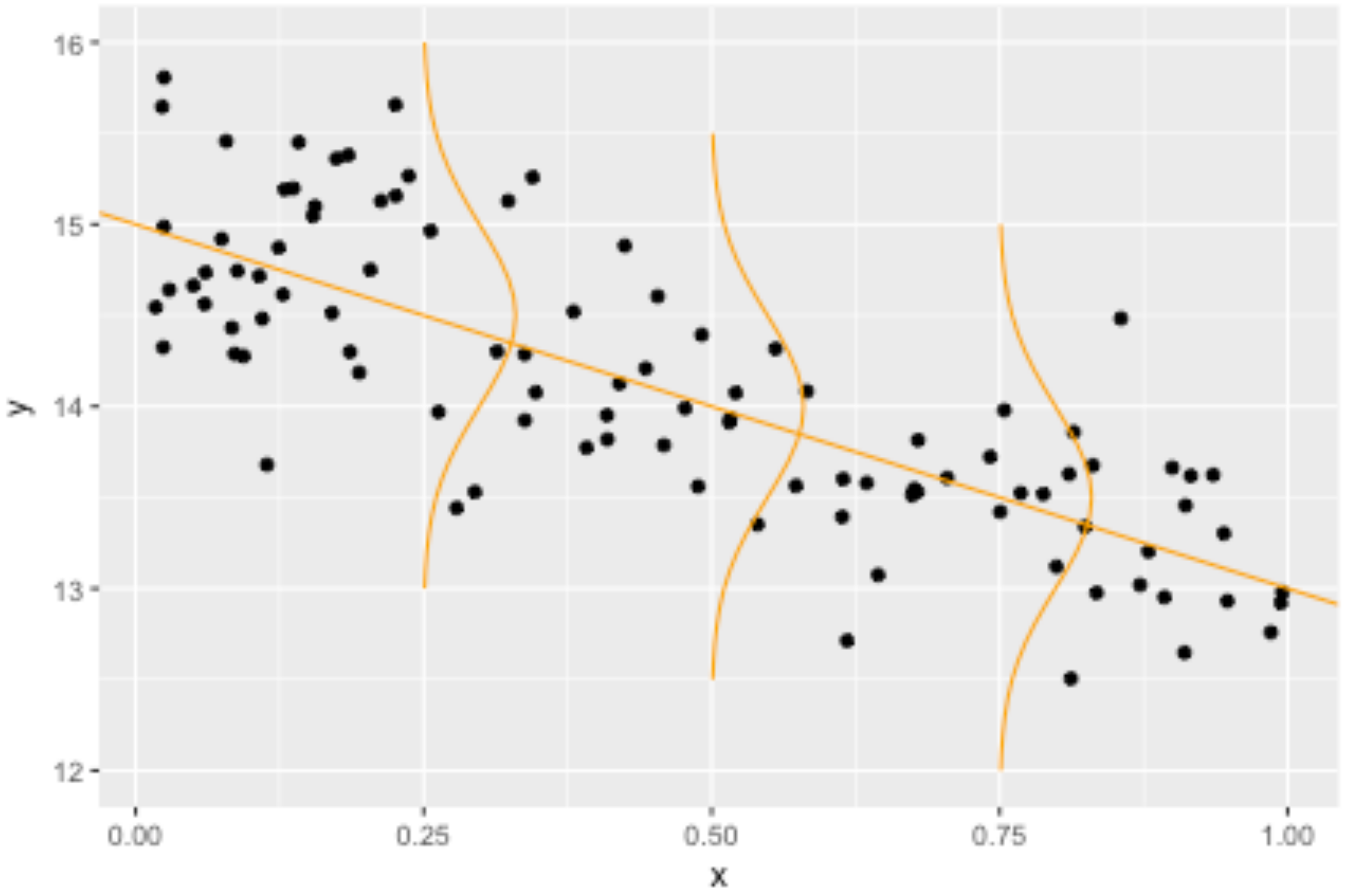
$$\hat{y} = b_0 + b_1 x_1 + \cdots + b_p x_p$$

Note the fitted model, and fitted values, are called *y hat* . The fitted values are points along the line.

Normal residuals

The assumption is that $\varepsilon \sim N(0, \sigma^2)$ implies that

$$y|x_1, \ldots, x_p \sim N(b_0 + b_1 x_1 + \cdots + b_p x_p, \sigma^2)$$



Regardless of the value of x the distribution of points above and below the line should be the same, and symmetric. And as you get further form the line, there should be less points.

Implications

- Optimisation function should consider the distribution assumption
- Need to check the residuals from the model fit satisfy the normality assumption.

Optimisation

Normal density function

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right\}$$

For error, $\mu = 0$.

The likelihood function is the product of the density function evaluated for each sample value, $x_1, \ldots, x_n$.

$$l(\mu, \sigma | x_1, \ldots, x_n) = \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \left(\frac{x_1-\mu}{\sigma} \right)^2 \right\} \times \ldots \times \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \left(\frac{x_n-\mu}{\sigma} \right)^2 \right\}$$

E.g. suppose $x_1 = 1, x_2 = -3$, then the likelihood is, assuming $\mu = 0$,

$$l(\sigma | x_1 = 1, x_2 = -3) = \frac{1}{2\pi} \exp\left\{ -\frac{1}{2} \frac{(1+9)}{\sigma^2} \right\}$$

which is a function in σ. Optimise this function to get the maximum likelihood estimate for σ.

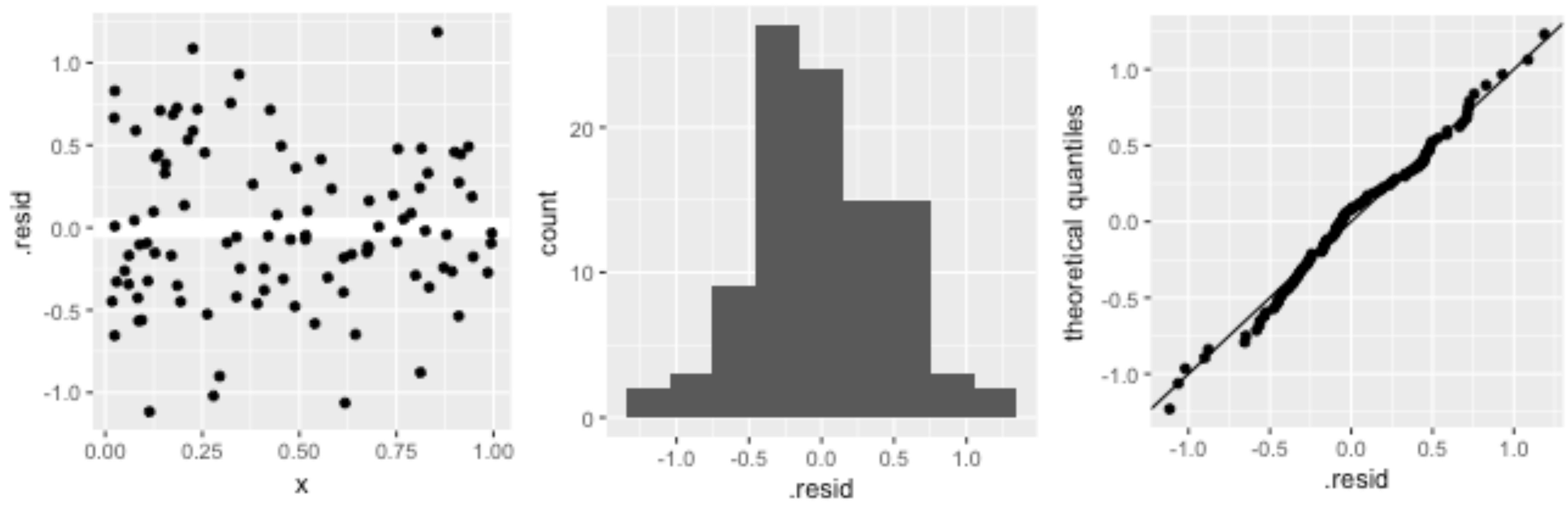
Intercept and slope

Set $\mu = \beta_0 + \cdots + \beta_1 x$ and then the likelihood is a function of $\sigma, \beta_0, \ldots, \beta_p$. Optimise the function over all of these parameters to get the maximum likelihood estimates for the linear model.

These will be the same as if you minimised the least squares equation, $\sum_i (y_i - (b_0 + b_1 x_1 + \cdots + b_p x_p))^2$.

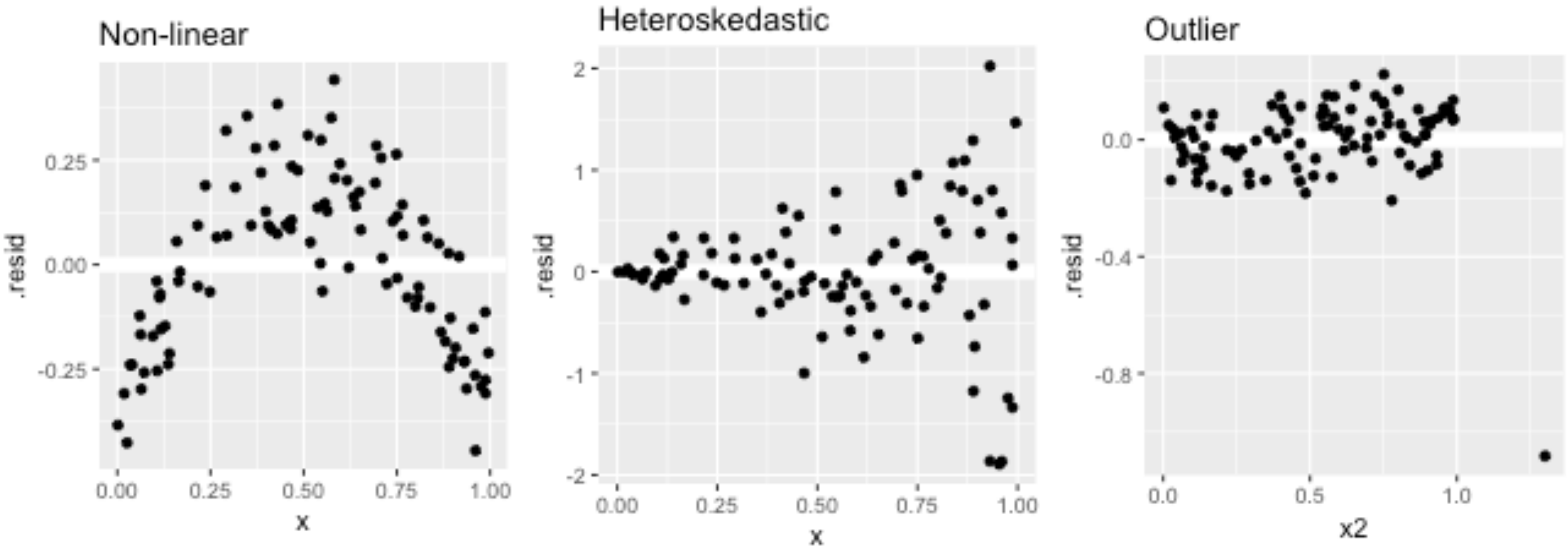
Checking normality

- Plot residuals vs fitted: is it nice and uniform?
- Make a histogram: is it symmetric, unimodal, no outliers?
- Make a normal probability plot: does it pass the fat marker test?

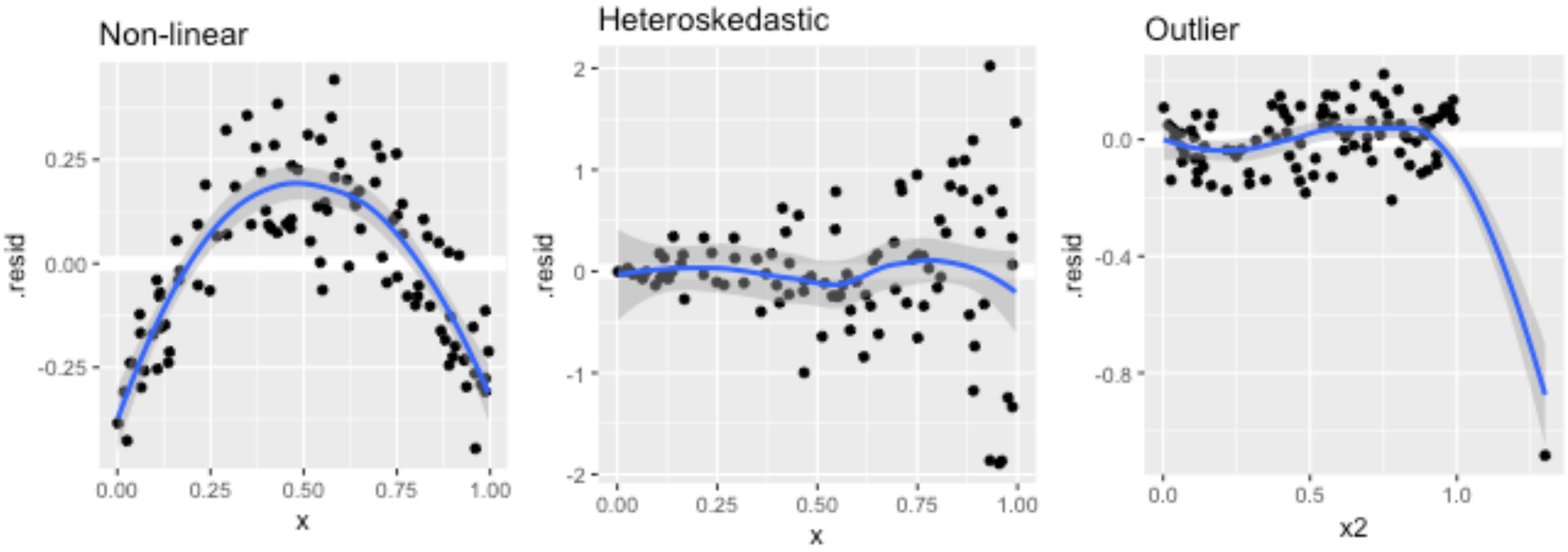


Common problems

- Non-linear relationship
- Heteroskedastic error
- Outliers



Its good to overlay a smoother sometimes.


Non-linear
.resid
0.25
0.00
-0.25
0.00
0.25
0.50
0.75
1.00
x
Heteroskedastic
.resid
2
1
0
-1
-2
0.00
0.25
0.50
0.75
1.00
x
Outlier
.resid
0.0
-0.4
-0.8
0.0
0.5
1.0
x2

Model fit

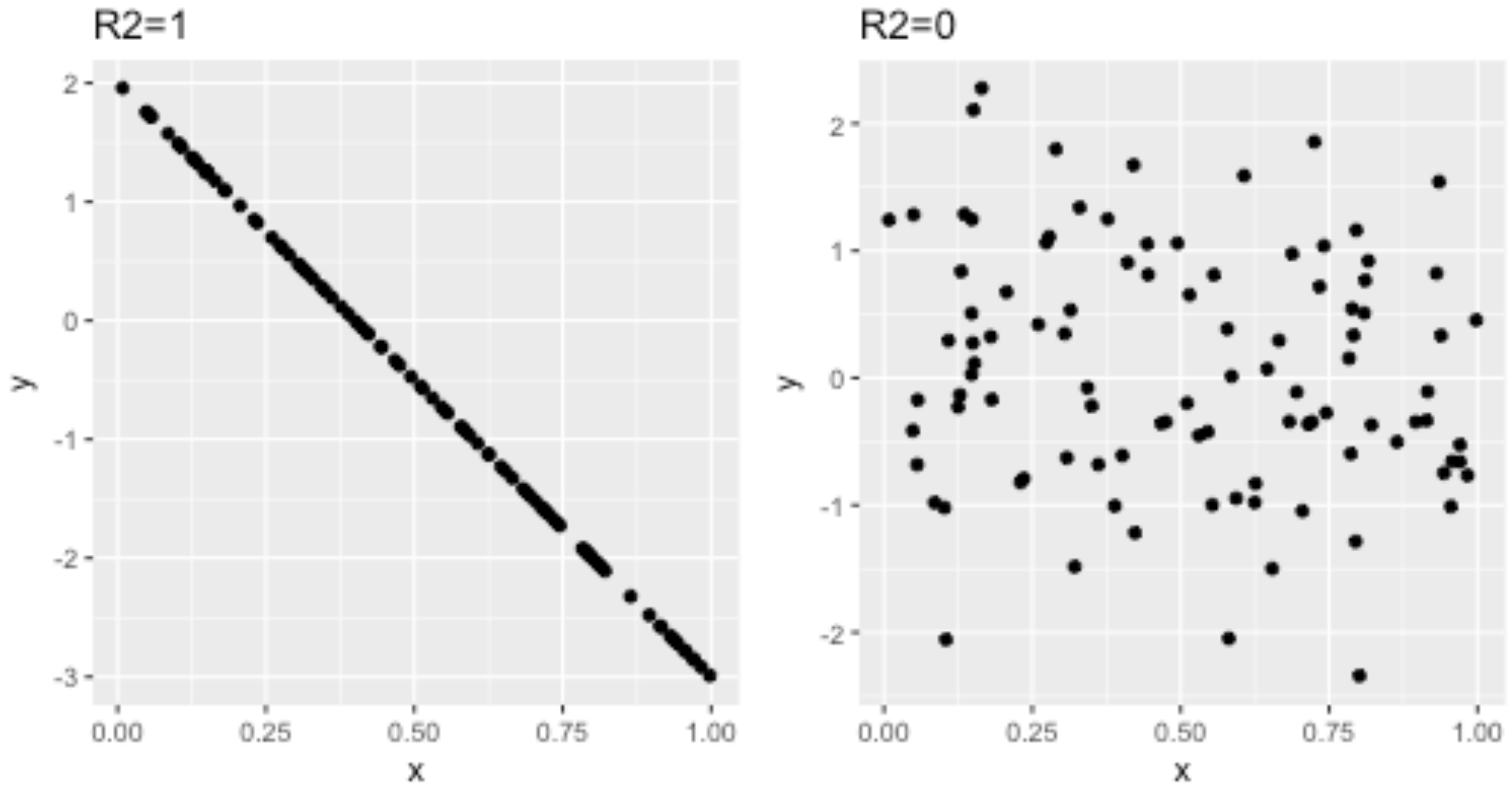
- R^2: proportion of variation explained by model
- deviance, null deviance
- AIC, BIC, negative loglikelihood

Proportion of variation explained

$$R^2 = 1 - \frac{SSE}{SST}$$

where $SST = \sum_i (y_i - \bar{y})^2$ and $SSE = \sum_i (y_i - \hat{y})^2$.

$0 < R^2 < 1$, where 1 would indicate the model explains ALL the variation in y, and 0 indicates it explains nothing.



Deviance and null deviance

- Most software does not report R^2 any more
- Related to the distributional assumptions on the error
- *deviance* : up to a constant, minus twice the maximized log-likelihood.
- *null deviance* : The deviance for the null model, comparable with deviance.
- A good model has a deviance that is much smaller than the null deviance, which means that it explains a lot of the variability in y. (Or the closer to 0 the better.)
- Deviance will decrease as more variables added to the model.

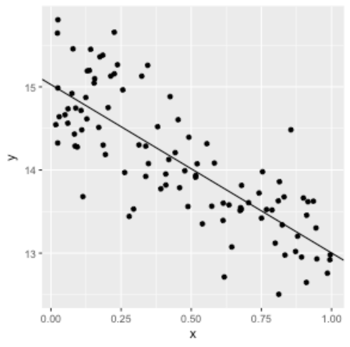
AIC, BIC

- AIC (Akaike Information Criterion): minus twice the maximized log-likelihood plus twice the number of parameters
- The lower the value the better the model
- Primarily used to compare models, pick the model with the lowest value
- BIC (Bayes Information Criterion) small variation, instead of twice, use `log(n)`, the number of parameters.

Model building

- Statistical tests can be used to determine whether parameter estimates indicate the true parameter is different from zero
- Use AIC to help select variables when there are many

Statistical tests on parameters



```
Call:
glm(formula = y ~ x, data = df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.11603  -0.31440  -0.06325   0.36795   1.18757

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 15.02681    0.08495  176.89   <2e-16 ***
x           -2.02429    0.15554  -13.02   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.225614)

    Null deviance: 60.324  on 99  degrees of freedom
Residual deviance: 22.110  on 98  degrees of freedom
AIC: 138.87

Number of Fisher Scoring iterations: 2
```

Statistical tests on parameters

- `t value` is calculated by `Estimate` divided by `Std. Error`
- If `t value` is large, indicates that the β_k (population parameter) is unlikely to be 0. Given what we have seen in the sample, it indicates that this parameter, and thus the variable associated with it is important (statistically significant) for explaining y.
- This should correspond to a `Pr(>|t|)` (*p-value*) being really small, less than 0.1.
- Both β_0 and β_1 in the example here are statistically significant (both different from 0) and so x_1 is really important for explaining y.

Interpretation

- Intercept: When $x_1 = 0$, then estimated y is b_0. Often doesn't make sense, but its important for the math to have this as part of the model
- Slope: For each unit increase in x_1, y increases, on average, by b_1.
- For multiple predictors, the interpretation of slope remains the same, assuming that all other variables are at fixed values.

Cautions

A model can be statistically significant but explain very little of the response variable. An example: Many restaurants in the USA have a policy *a tip rate of 18% will be charged to dining parties of six or more*. This comes from a linear model, like this:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  18.3191     2.0750   8.829 2.44e-16 ***
size         -0.9625     0.4217  -2.282   0.0234 *
sexMale      -0.8543     0.8348  -1.023   0.3072
smokerYes     0.3637     0.8497   0.428   0.6690
daySat       -0.1773     1.8341  -0.097   0.9231
daySun        1.6672     1.9023   0.876   0.3817
dayThur      -1.8176     2.3194  -0.784   0.4340
timeLunch     2.3371     2.6118   0.895   0.3718
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 36.79008)

    Null deviance: 9063.4  on 243  degrees of freedom
Residual deviance: 8682.5  on 236  degrees of freedom
AIC: 1582

Number of Fisher Scoring iterations: 2
```

Interpretation

The only important variable in the model is `Size`.

$$\text{Tip percentage} = 18.3 - 0.96 \times \text{Size of the dining party}$$

For each additional member in the dining party, the tip % decreases by about 1%.

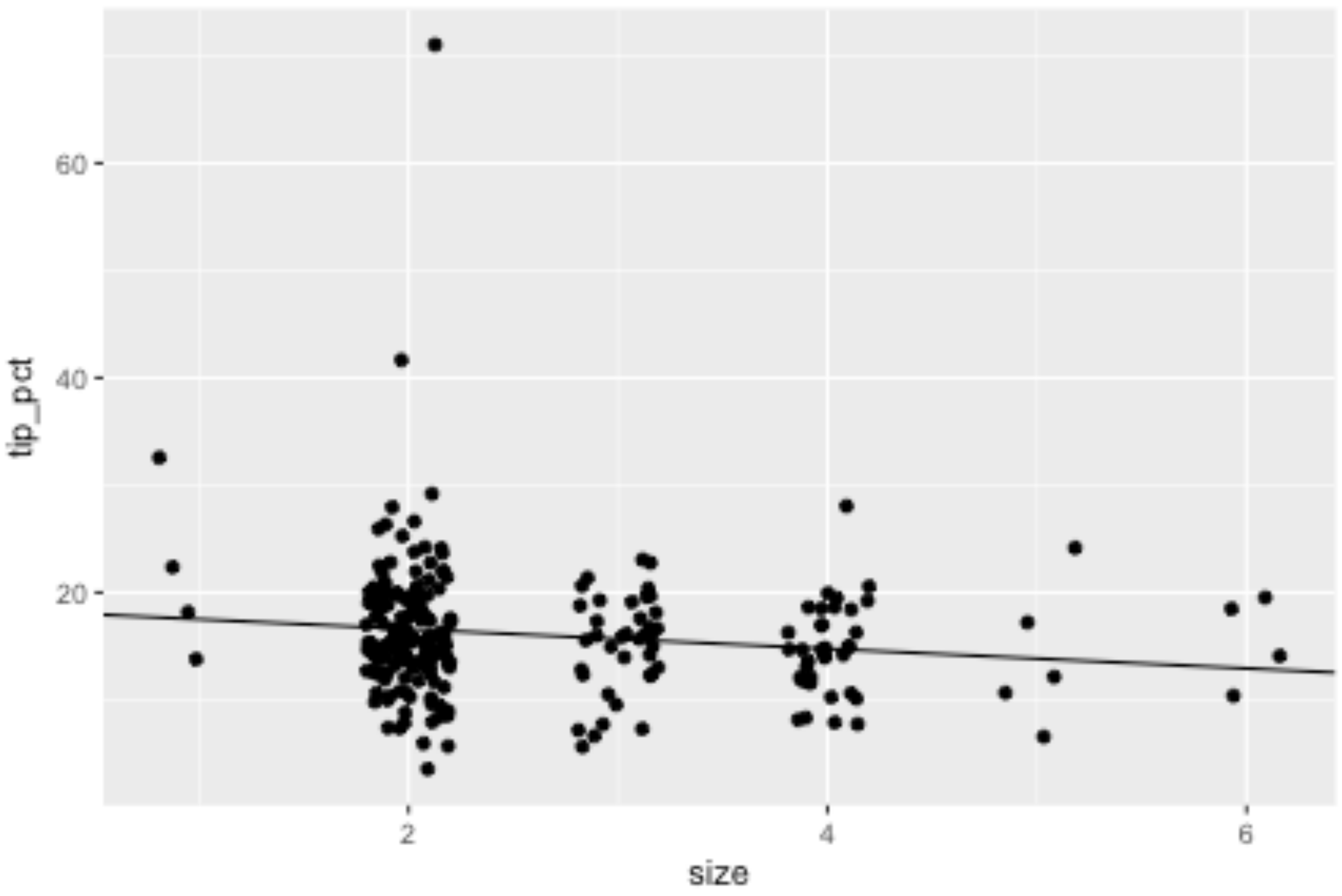
But look at deviance (8682.5) relative to the null deviance (9063.4). There is very little difference between the two, which means that size of the dining party explains very little of the variation in tip percentage.

If we prefer to use R^2 use the `lm` function instead:

```
Call:
lm(formula = tip_pct ~ size, data = tips)

Residuals:
    Min      1Q  Median      3Q     Max 
-13.039  -3.077  -0.608   3.148  54.432 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  18.4375     1.1191  16.475   <2e-16 ***
size         -0.9173     0.4085  -2.245   0.0256 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.057 on 242 degrees of freedom
Multiple R-squared:  0.02041,    Adjusted R-squared:  0.01636 
F-statistic: 5.042 on 1 and 242 DF,  p-value: 0.02565
```

The model is statistically significant but it explains only 2% of the variation in tip percentage. It is practically useless. There are a lot of other factors that affect tips. However, restauranteurs have picked up on the relationship and instituted a convenient policy to guarantee a minimal tip intake with larger dining parties.

Make a plot



Regression trees

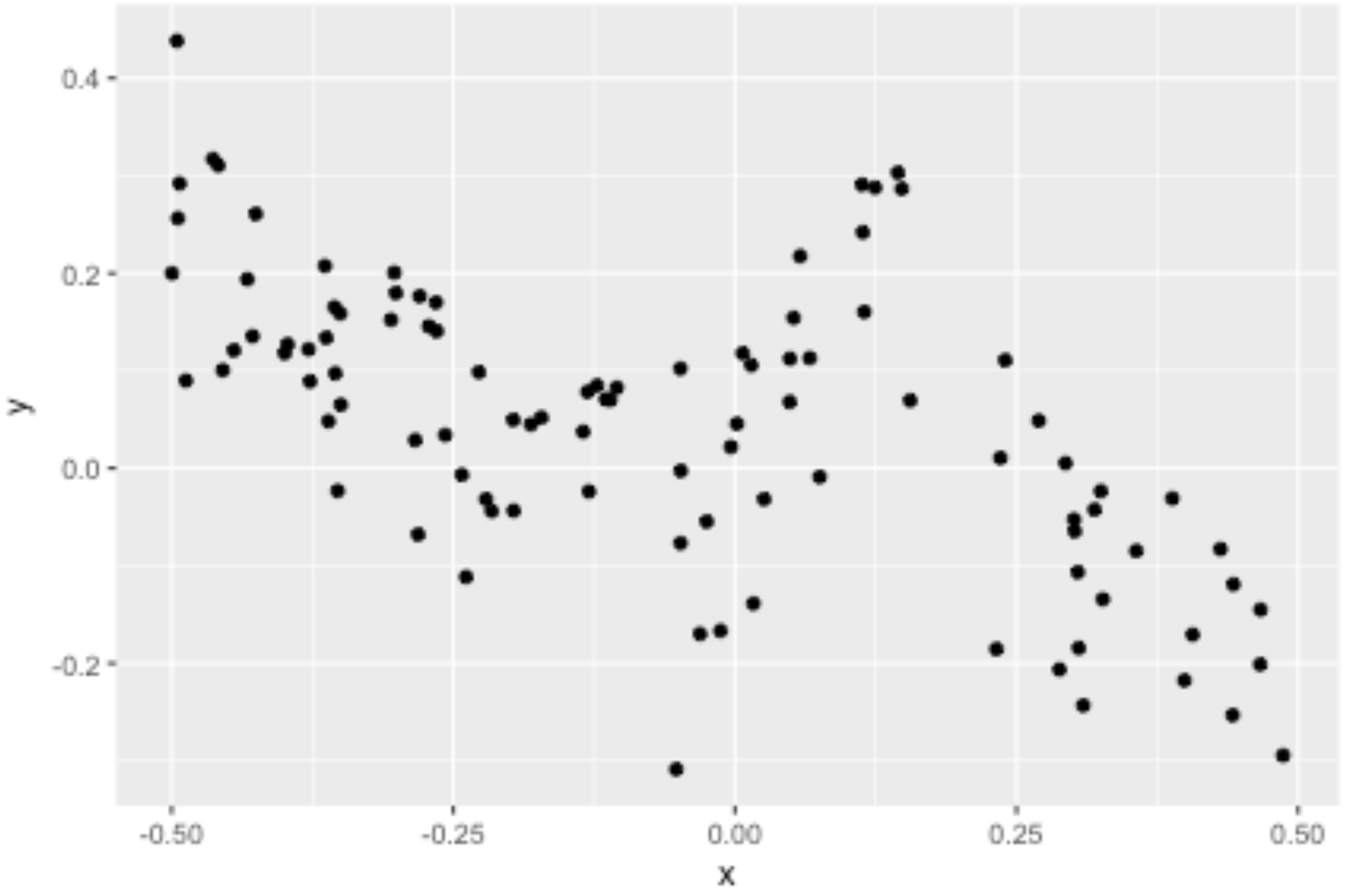
- Regression (decision) trees recursively partition the data, and use the average response value of each partition as the model estimate
- Computationally intensive technique, involves examining ALL POSSIBLE partitions.
- Chooses the BEST partition by optimizing a criteria
- For regression, with a quantitative response variable, the criteria is called ANOVA:

$$SS_T - (SS_L + SS_R)$$

where $SS_T = \sum(y_i - \bar{y})^2$, and SS_L, SS_R are the equivalent values for the two subsets created by partitioning.

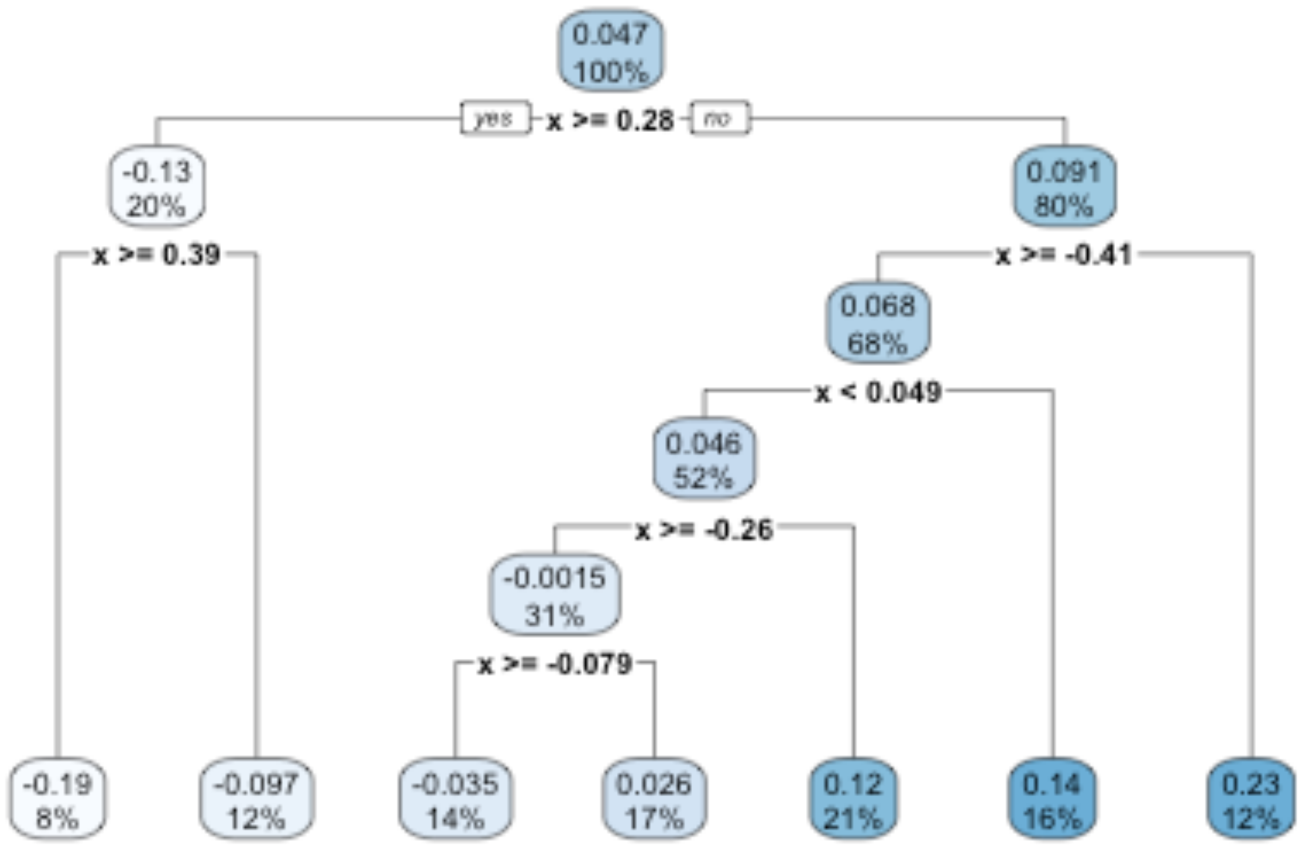
What it looks like

Here's a synthetic data set for illustration



```
df_rp <- rpart(y~x, data=df)
df_rp
n= 100

node), split, n, deviance, yval
      * denotes terminal node

 1) root 100 2.26392100  0.046658990
   2) x>=0.2789237 20 0.14061080 -0.132608100
     4) x>=0.3936632 8 0.03452376 -0.185497100 *
     5) x< 0.3936632 12 0.06879035 -0.097348760 *
   3) x< 0.2789237 80 1.31989300  0.091475770
     6) x>=-0.4125869 68 0.94100140  0.067680920
      12) x< 0.04880311 52 0.56644400  0.045942920
        24) x>=-0.2610339 31 0.29105710 -0.001481586
          48) x>=-0.07853794 14 0.20666590 -0.034887260 *
          49) x< -0.07853794 17 0.05590193  0.026028970 *
        25) x< -0.2610339 21 0.10274290  0.115950500 *
      13) x>=0.04880311 16 0.27012590  0.138329400 *
     7) x< -0.4125869 12 0.12221610  0.226313300 *
```

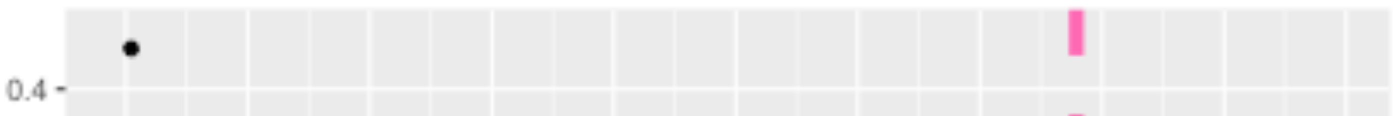
Model decision tree

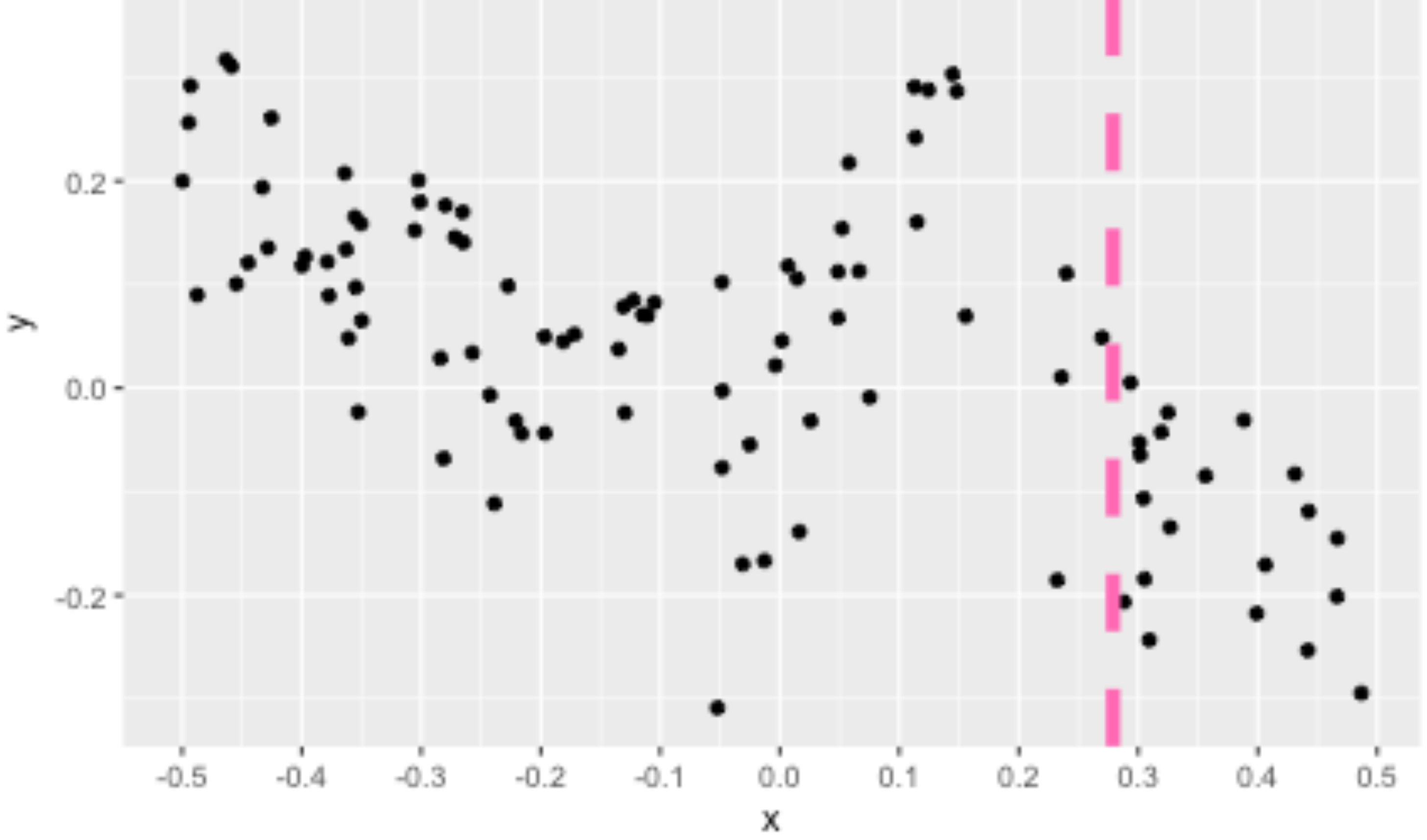


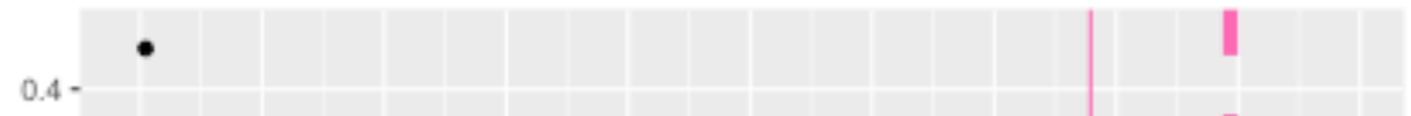
Next, picture the model on the data

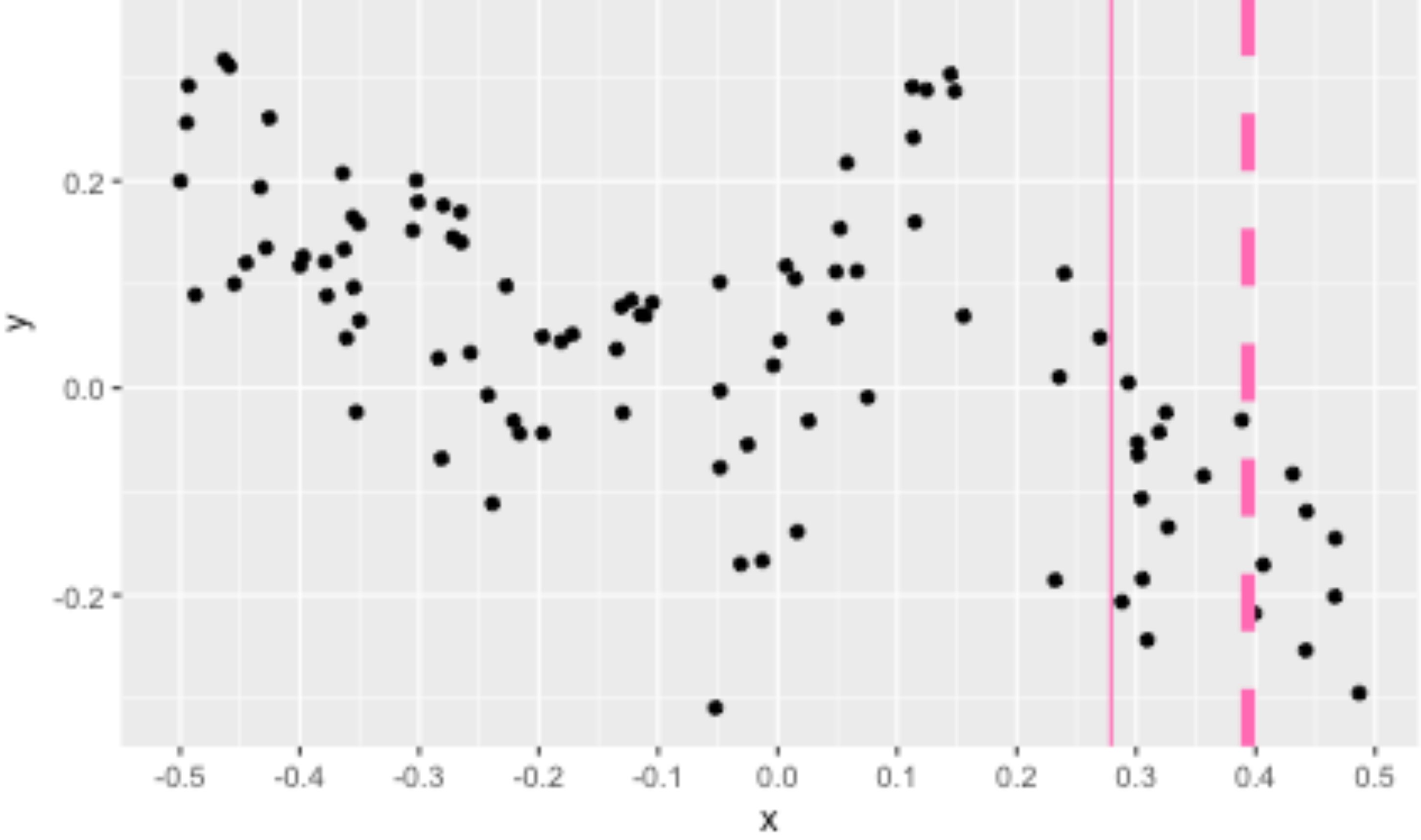
Splits

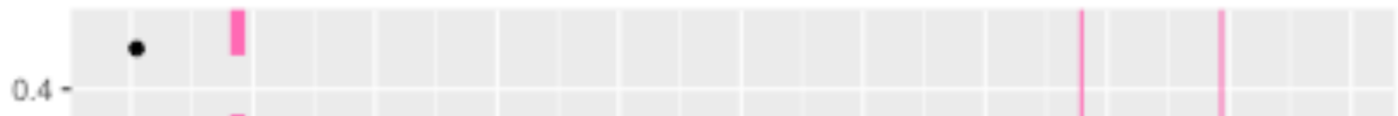
```
  count ncat    improve       index adj
x   100    1 0.35487877  0.27892366   0
x    20    1 0.26524753  0.39366322   0
x    80    1 0.19446662 -0.41258686   0
x    68   -1 0.11097910  0.04880311   0
x    52    1 0.30478553 -0.26103391   0
x    31    1 0.09788205 -0.07853794   0
```

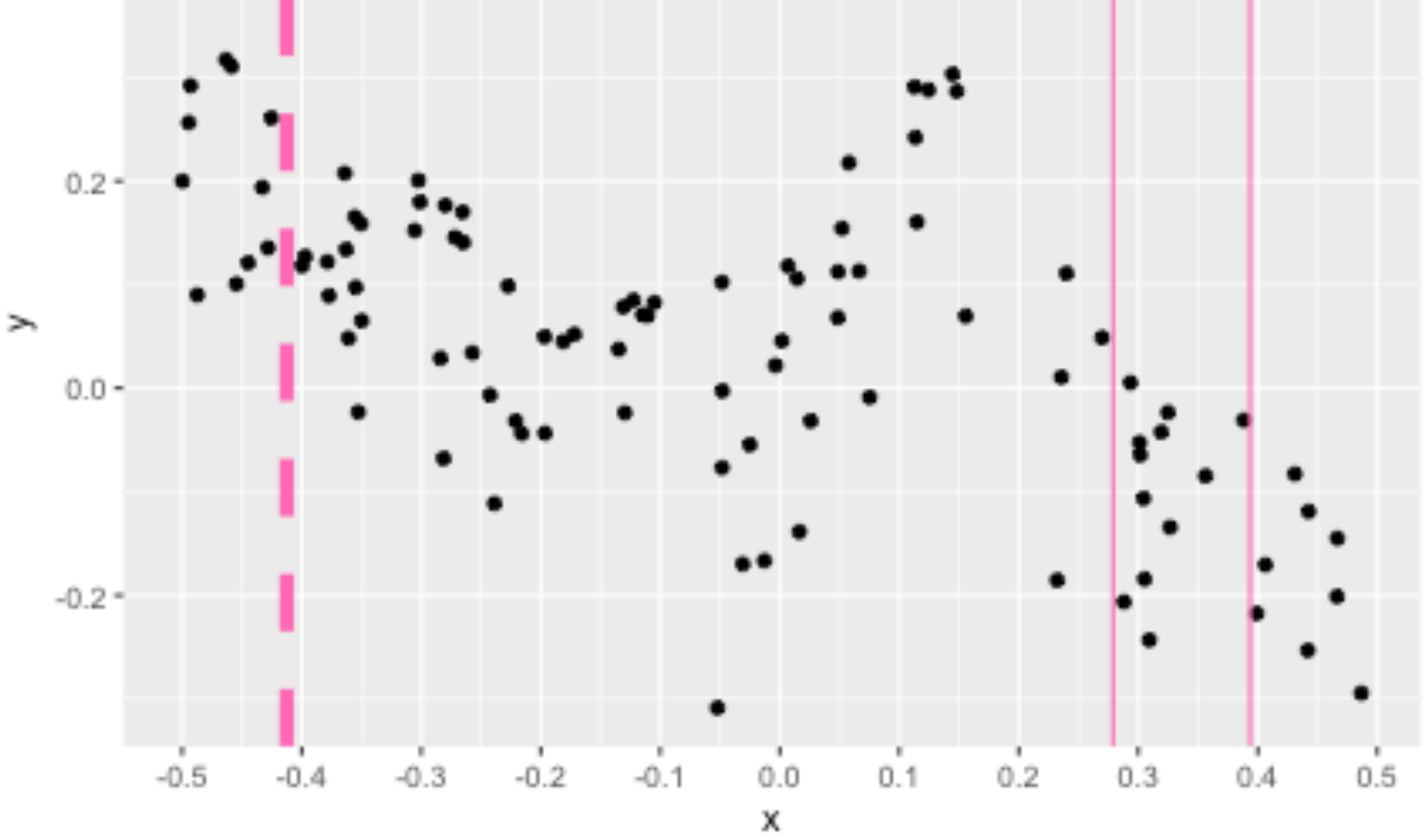


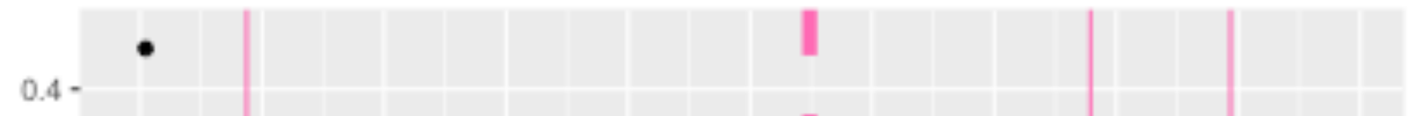

y
0.2
0.0
-0.2
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
x

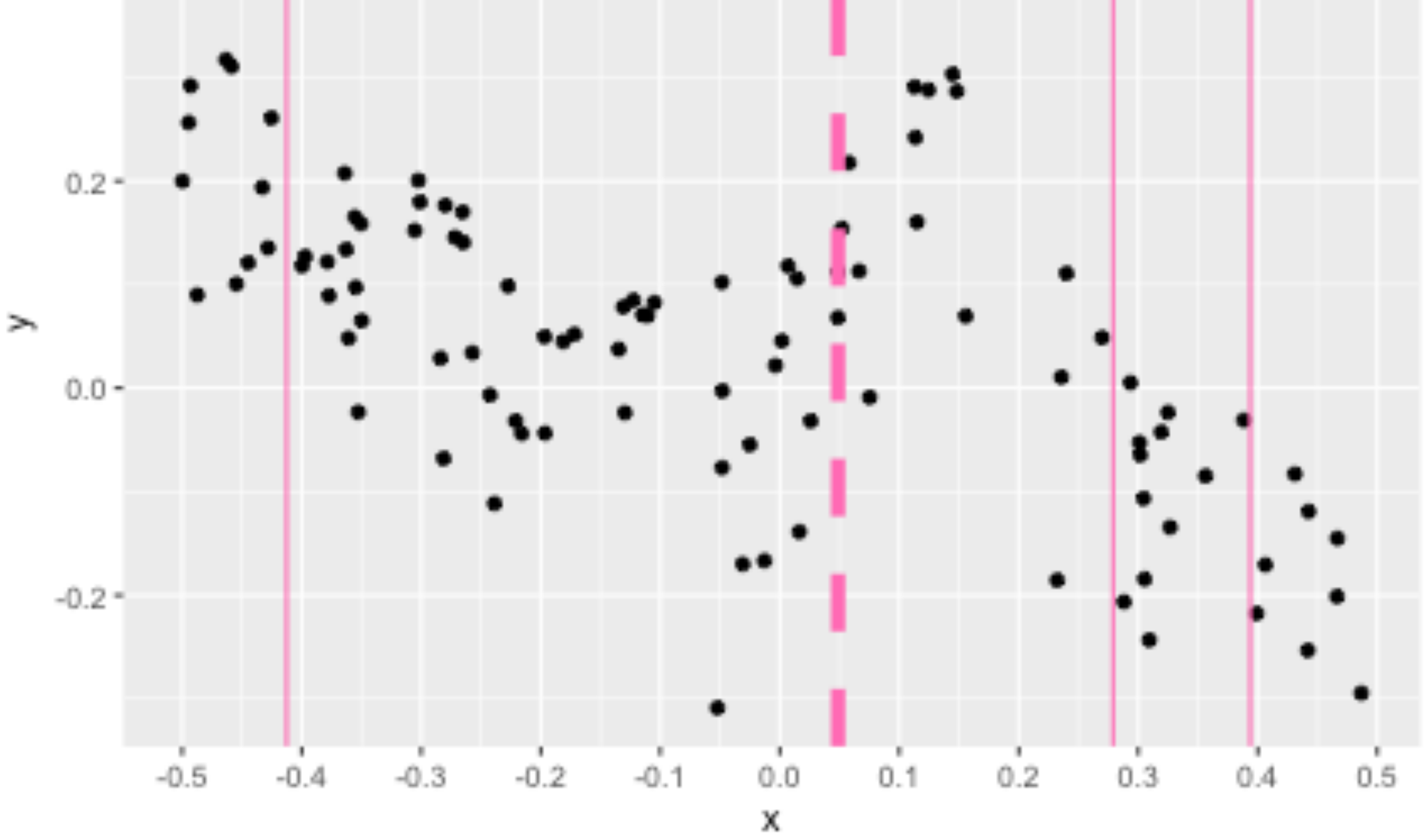

0.4

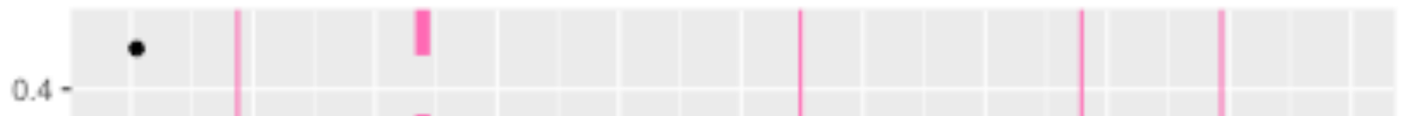

0.2
0.0
-0.2
y
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
x

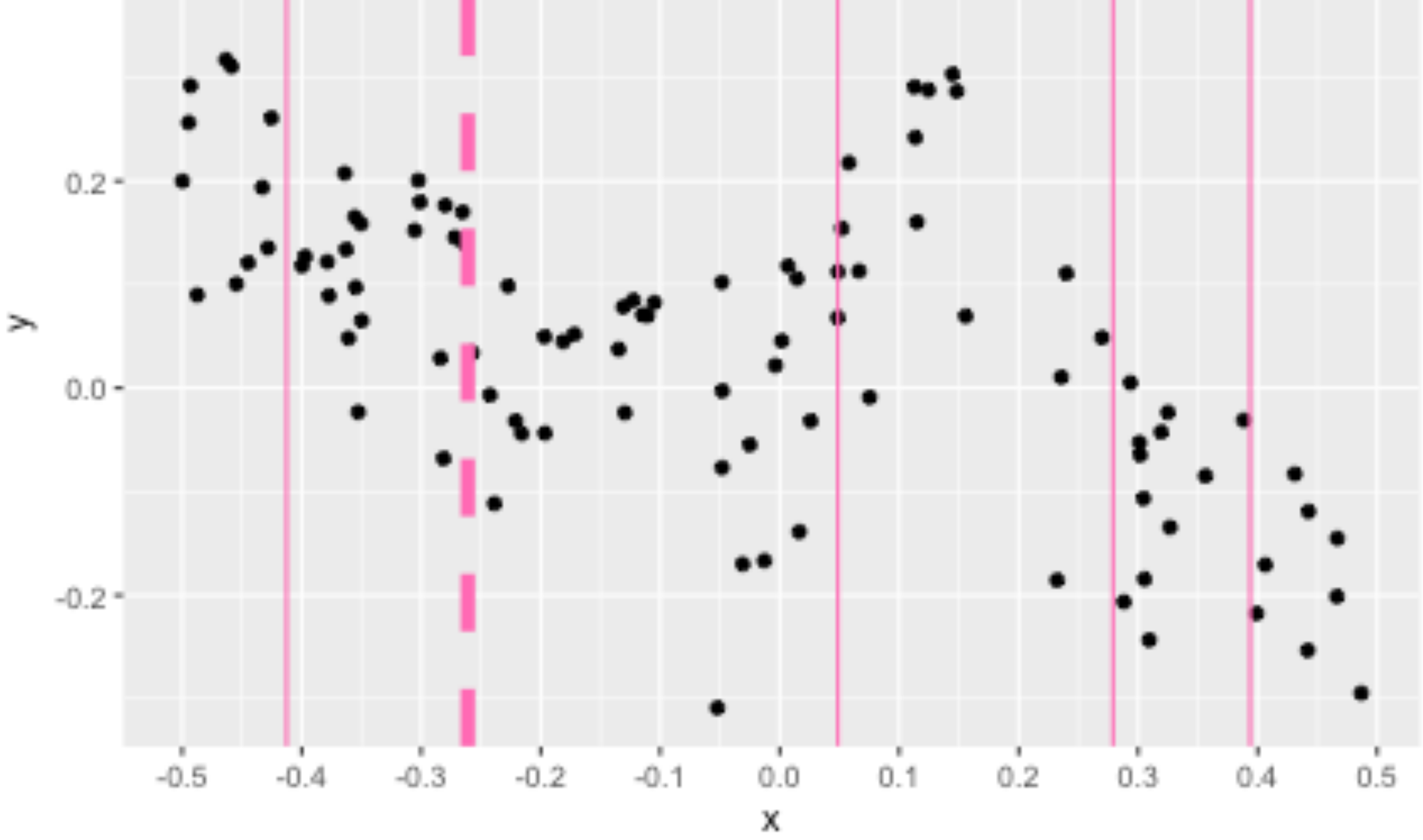

0.4

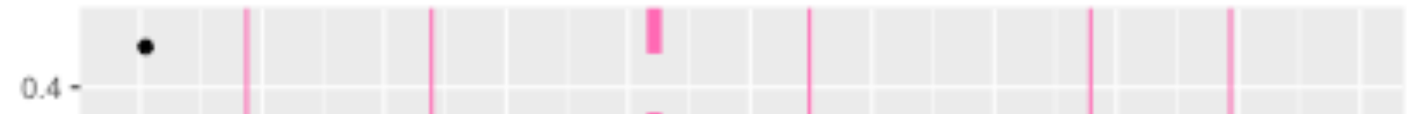

0.2
0.0
-0.2
y
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
x

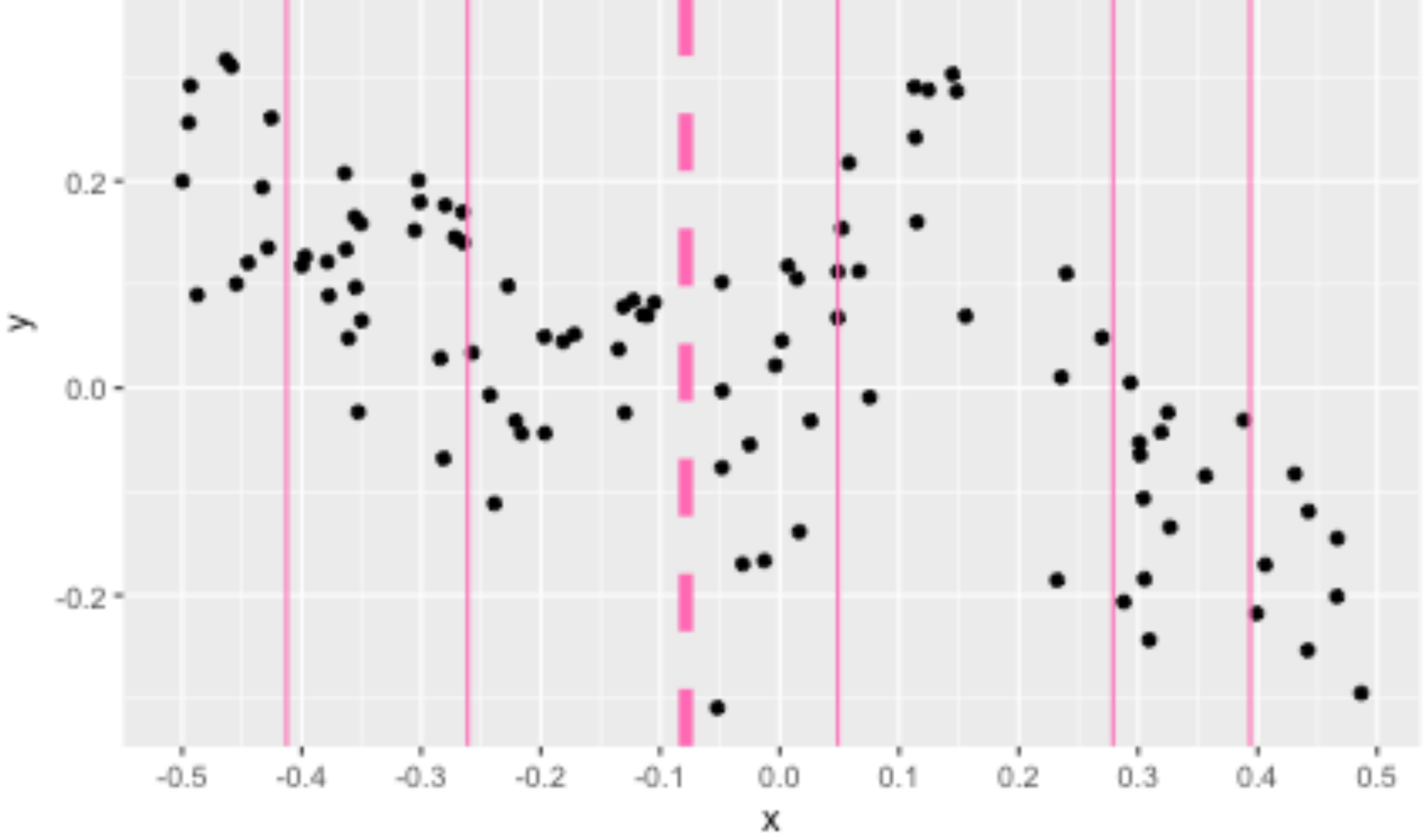

0.4

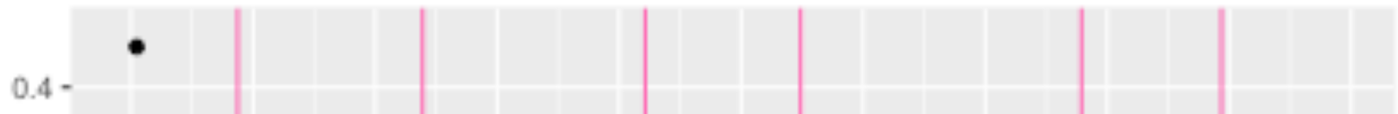

y
0.2
0.0
-0.2
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
x

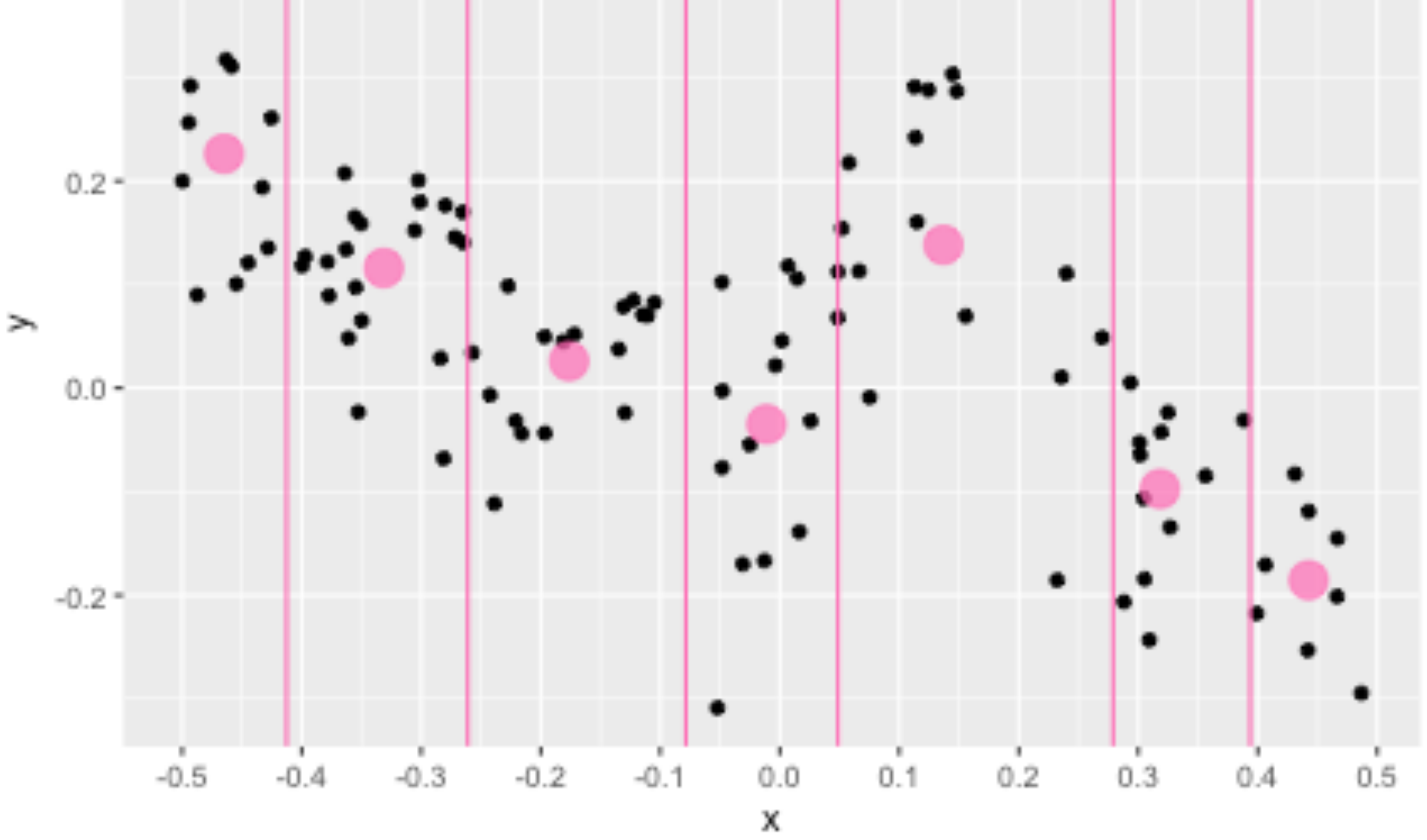

0.4


0.2
0.0
-0.2
y
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
x


0.4


0.2
0.0
-0.2
y
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
x


0.4


0.2
0.0
-0.2
y
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
x

When do we stop?

- Its an algorithm. Why did it stop at 7 groups?
- Stopping rules ar needed, else the algorithm will keep fitting until every observartion is in its own group.
- Control parameters set stopping points:
  - minsplit: minimum number of points in a node that algorithm is allowed to split
  - minbucket: minimum number of points in a terminal node
- In addition, we can also look at the change in value of $SS_T - (SS_L + SS_R)$ at each split, and if the change is too *small*, stop. To decide on a suitable value for *small* a cross-validation procedure is used.

Stop points in example model

```
List of 9
 $ minsplit      : int 20
 $ minbucket     : num 7
 $ cp            : num 0.01
 $ maxcompete    : int 4
 $ maxsurrogate  : int 5
 $ usesurrogate  : int 2
 $ surrogatestyle: int 0
 $ maxdepth      : int 30
 $ xval          : int 10
```

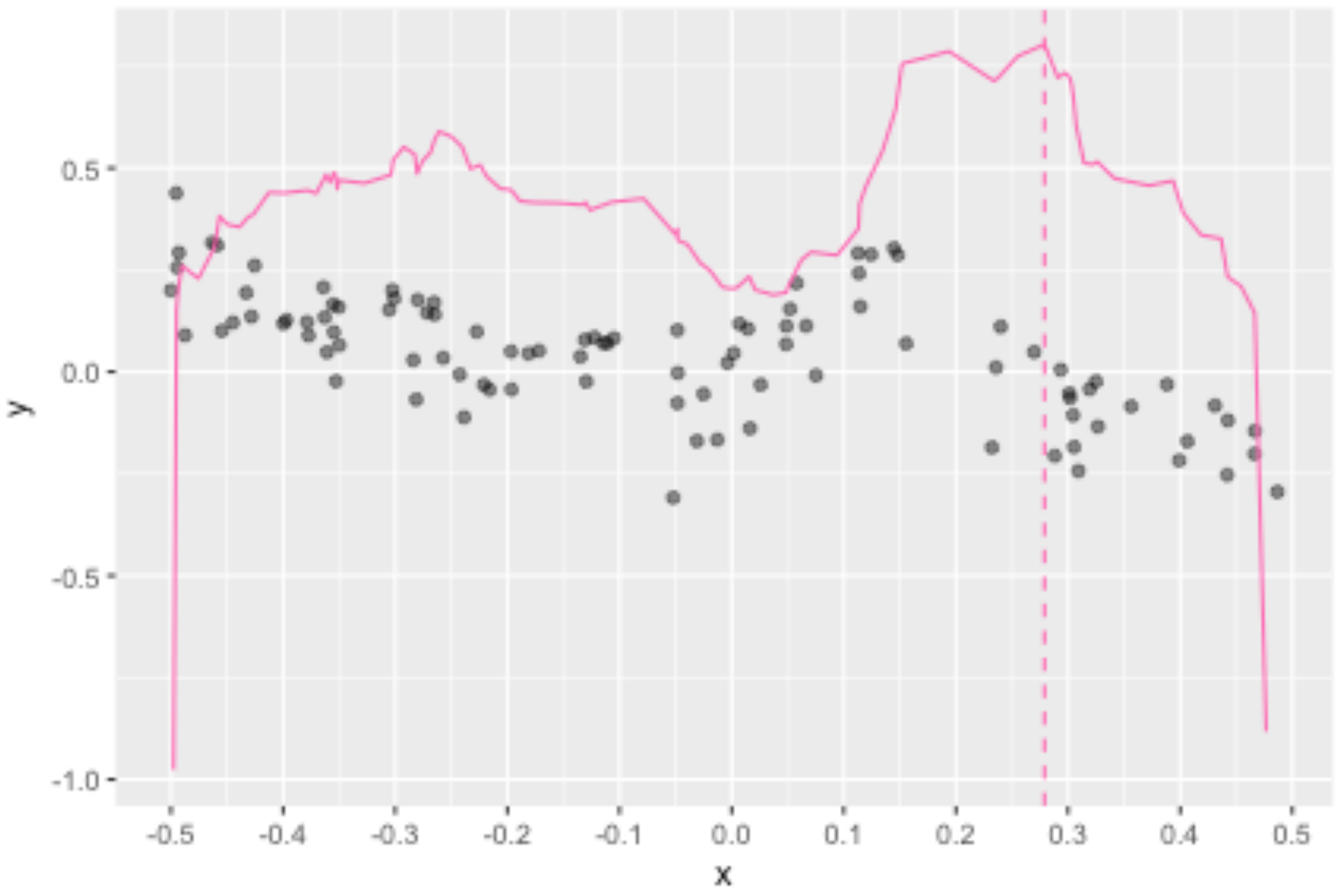
Changing control parameters

```
df_rp <- rpart(y~x, data=df,
  control = rpart.control(minsplit=5, minbucket = 2))
df_rp
n= 100

node), split, n, deviance, yval
      * denotes terminal node

 1) root 100 2.26392100  0.046658990
   2) x>=0.2789237 20 0.14061080 -0.132608100
     4) x>=0.3936632 8 0.03452376 -0.185497100 *
     5) x< 0.3936632 12 0.06879035 -0.097348760 *
   3) x< 0.2789237 80 1.31989300  0.091475770
     6) x>=-0.4125869 68 0.94100140  0.067680920
      12) x< 0.04880311 52 0.56644400  0.045942920
        24) x>=-0.2610339 31 0.29105710 -0.001481586
          48) x>=-0.07853794 14 0.20666590 -0.034887260
            96) x< -0.008211745 7 0.10571210 -0.096719830 *
            97) x>=-0.008211745 7 0.04742810  0.026945310 *
          49) x< -0.07853794 17 0.05590193  0.026028970 *
        25) x< -0.2610339 21 0.10274290  0.115950500 *
      13) x>=0.04880311 16 0.27012590  0.138329400
        26) x>=0.1518302 5 0.05338438  0.010833720 *
        27) x< 0.1518302 11 0.09852226  0.196282000
          54) x< 0.09397898 5 0.02736176  0.117667800 *
          55) x>=0.09397898 6 0.01450874  0.261793800 *
     7) x< -0.4125869 12 0.12221610  0.226313300
```

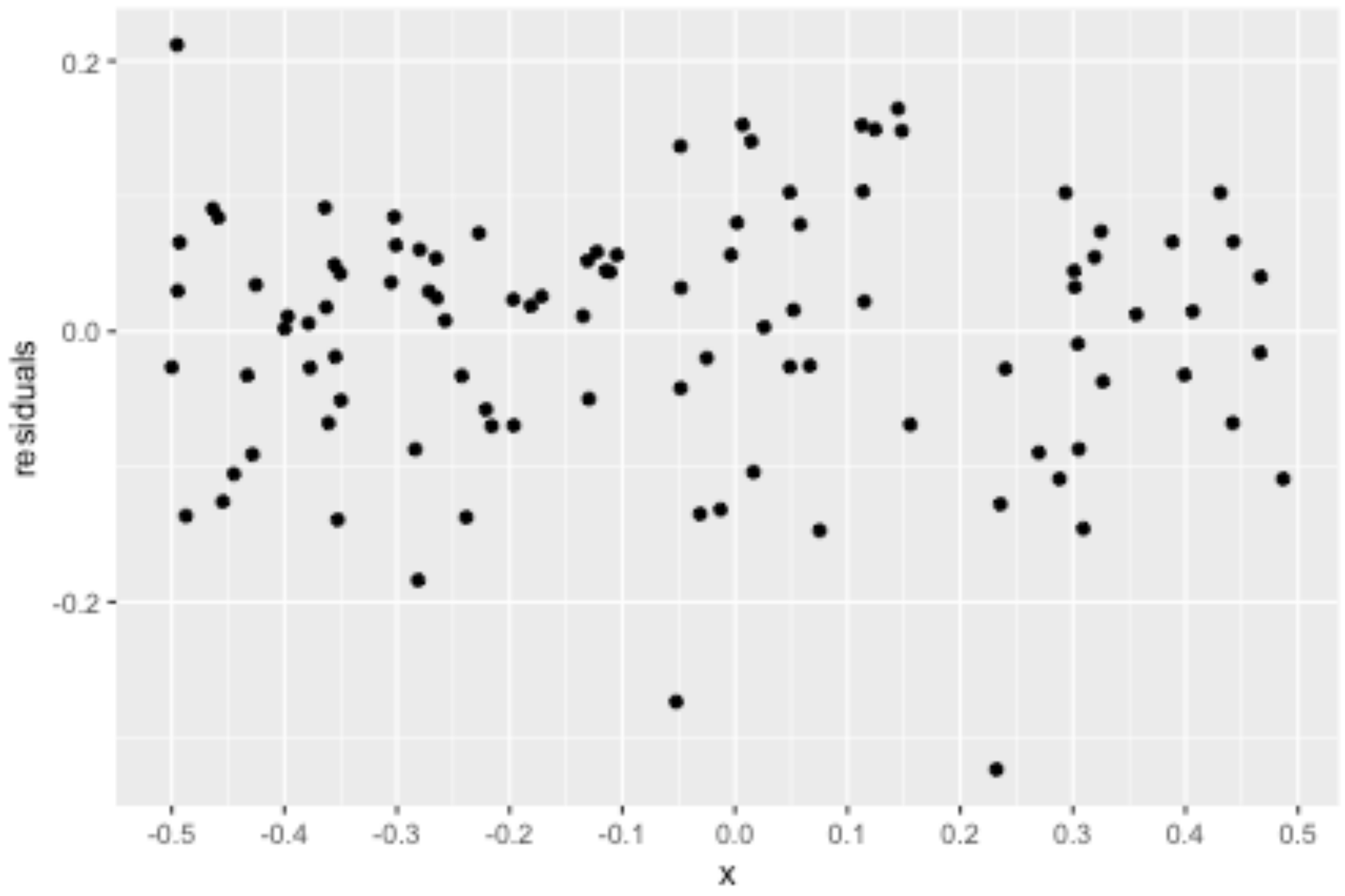
```
df_rp <- rpart(y~x, data=df,
  control = rpart.control(minsplit=20, minbucket = 10))
  control = rpart.control(minsplit=30, minbucket = 10))
df_rp
n= 100

node), split, n, deviance, yval
      * denotes terminal node

 1) root 100 2.26392100  0.046658990
   2) x>=0.2789237 20 0.14061080 -0.132608100 *
   3) x< 0.2789237 80 1.31989300  0.091475770
     6) x>=-0.4125869 68 0.94100140  0.067680920
      12) x< 0.04880311 52 0.56644400  0.045942920
        24) x>=-0.2610339 31 0.29105710 -0.001481586
          48) x>=-0.07853794 14 0.20666590 -0.034887260 *
          49) x< -0.07853794 17 0.05590193  0.026028970 *
        25) x< -0.2610339 21 0.10274290  0.115950500 *
      13) x>=0.04880311 16 0.27012590  0.138329400 *
     7) x< -0.4125869 12 0.12221610  0.226313300 *
```

What's the computation?

Illustration showing the calculations made to decide on the first partition.



Residuals



Goodness of fit

```
gof <- printcp(df_rp, digits=3)

Regression tree:
rpart(formula = y ~ x, data = df)

Variables actually used in tree construction:
[1] x

Root node error: 2.26/100 = 0.0226

n= 100

      CP nsplit rel error xerror   xstd
1 0.3549      0     1.000  1.022 0.1341
2 0.1134      1     0.645  0.822 0.1193
3 0.0612      2     0.532  0.682 0.0931
4 0.0165      4     0.409  0.559 0.0807
5 0.0126      5     0.393  0.539 0.0737
6 0.0100      6     0.380  0.536 0.0738
```

The relative error is $1 - R^2$. For this example, after 6 splits it is 0.3802991. So $R^2 =$ 0.6197009.

Strengths and weaknesses

- There are no parametric assumptions underlying partitioning methods
- Also means that there is not a nice formula for the model as a result, or inference about populations available
- By minimizing sum of squares (ANOVA) we are forcing the partitions to have relatively equal variance. The method could be influenced by outliers, but it would be isolating the effect to one partition.
- Because it operates on single variables, it can efficiently handle missing values.

Example

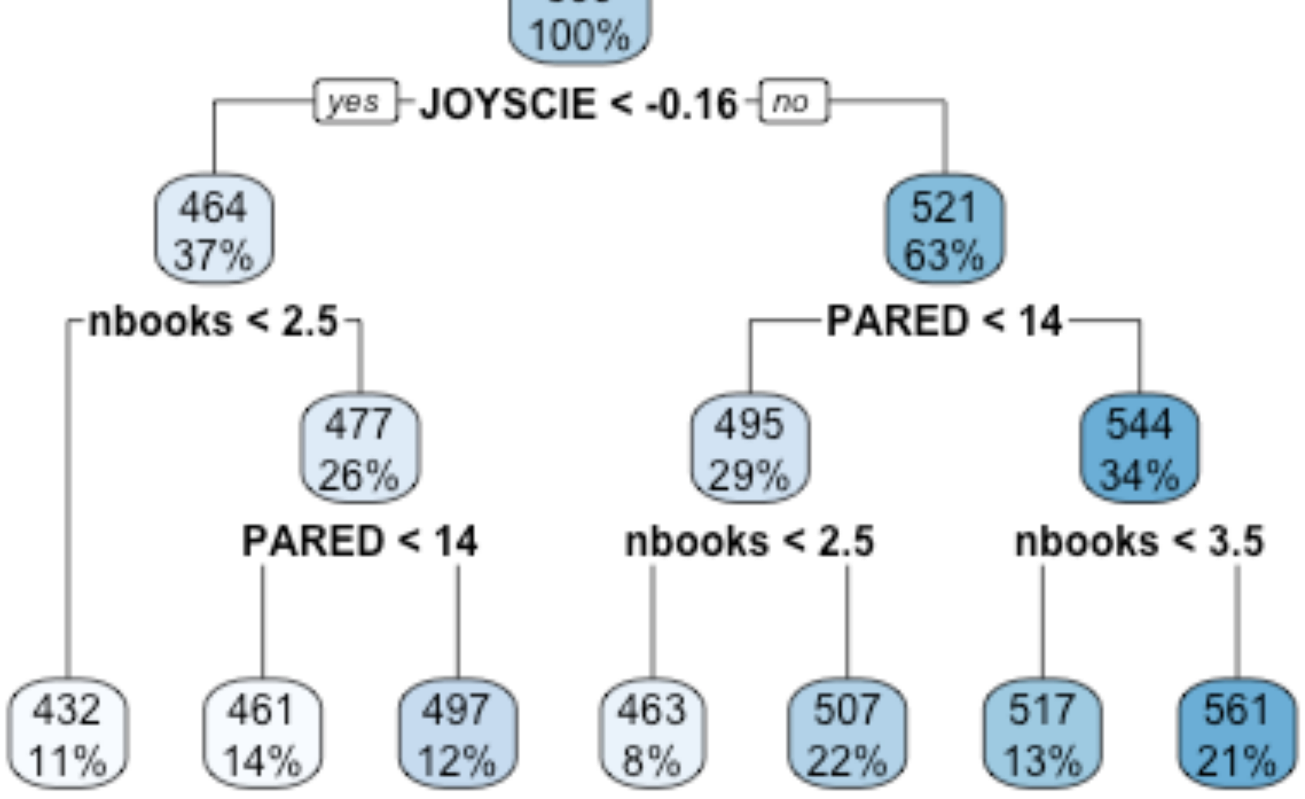
- OECD PISA, what factors affect reading scores?
- 15 year old standardised test scores, Australia, 2015
- Response: math
- Predictors: gender, ANXTEST, PARED, JOYSCIE, WEALTH, nbooks, ntvs

Linear model

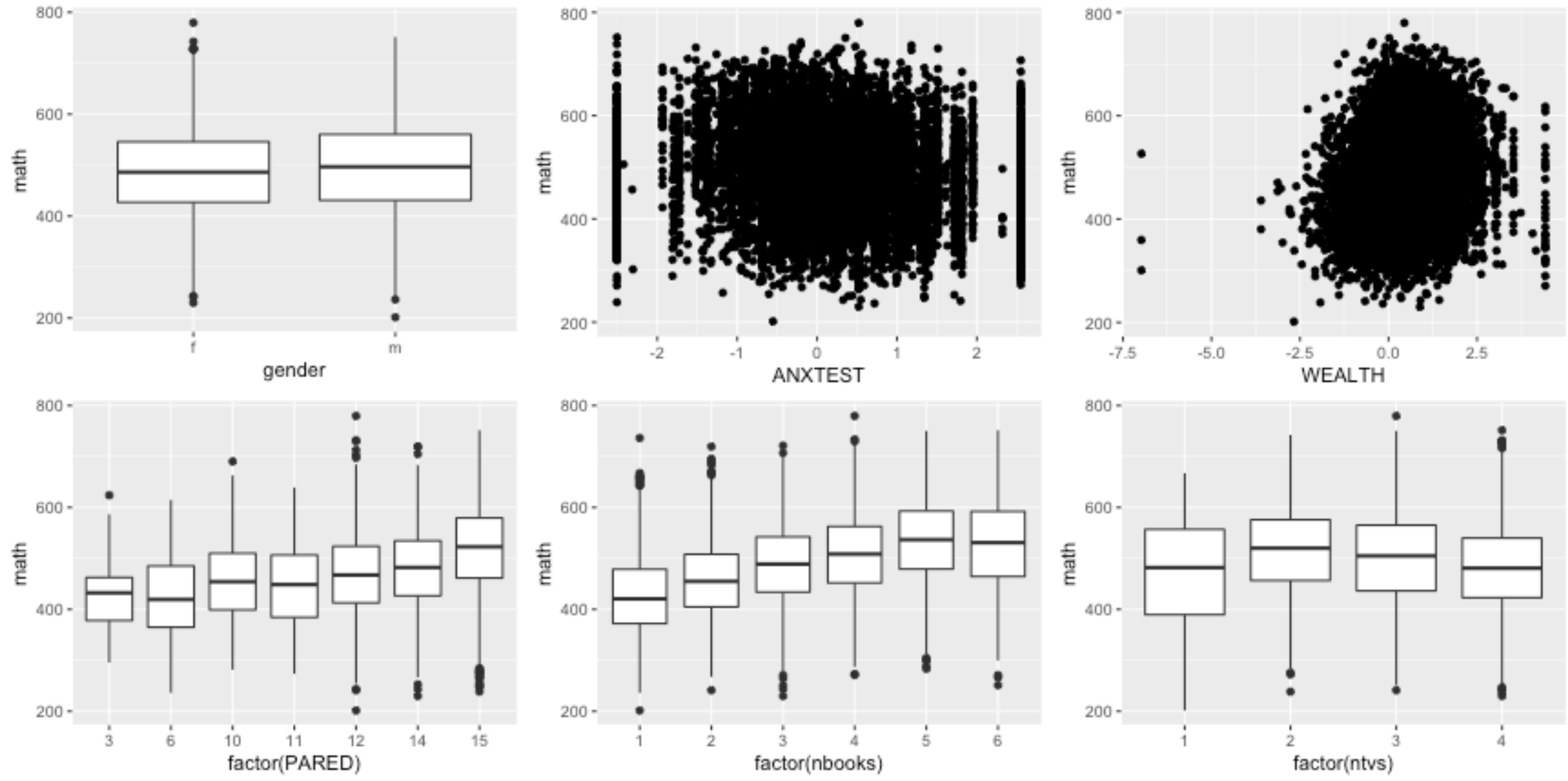
```
Call:
lm(formula = math ~ gender + ANXTEST + PARED + JOYSCIE + WEALTH +
    nbooks + ntvs, data = pisa_au_nomiss, weights = W_FSTUWT)

Weighted Residuals:
     Min       1Q   Median       3Q      Max 
-1535.64  -179.23   -11.07   168.42  1336.37 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 388.1974     6.3093  61.527  < 2e-16 ***
genderm       5.1320     1.3661   3.757 0.000173 ***
ANXTEST      -8.0380     0.6956 -11.556  < 2e-16 ***
PARED         7.1283     0.3603  19.787  < 2e-16 ***
JOYSCIE      21.2261     0.5740  36.981  < 2e-16 ***
WEALTH        5.6973     0.8923   6.385 1.78e-10 ***
nbooks       14.4634     0.4949  29.226  < 2e-16 ***
ntvs        -12.1794     1.0491 -11.609  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 308 on 12110 degrees of freedom
Multiple R-squared:  0.2739,	Adjusted R-squared:  0.2735 
F-statistic: 652.6 on 7 and 12110 DF,  p-value: < 2.2e-16
```

Regression tree

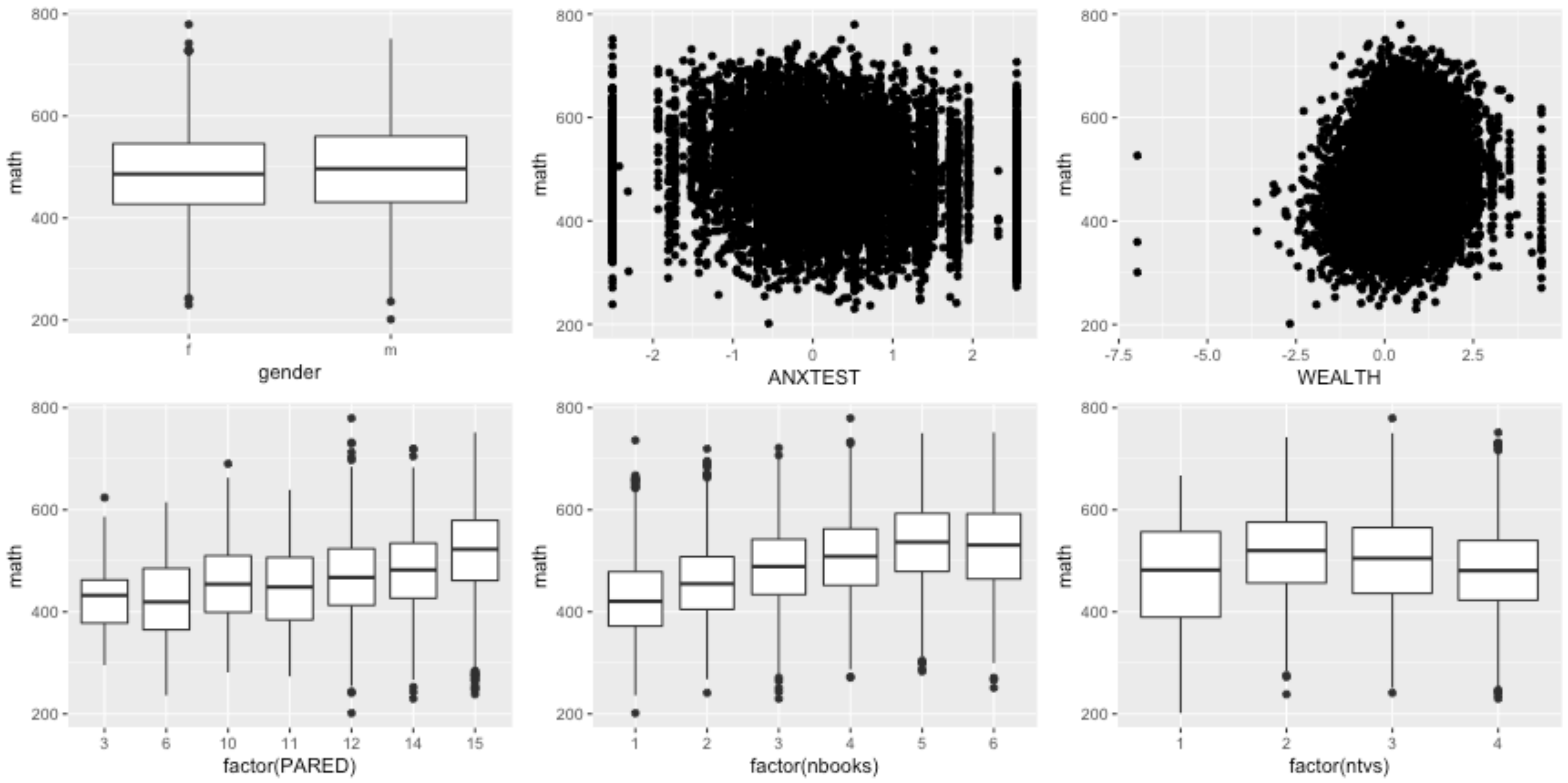
```
n= 12118

node), split, n, deviance, yval
      * denotes terminal node

 1) root 12118 1581871000 500.3926
   2) JOYSCIE< -0.1626 4631  465314500 464.3309
     4) nbooks< 2.5 1512  117372100 432.4937 *
     5) nbooks>=2.5 3119  314200900 477.2861
      10) PARED< 14.5 1755  146736900 461.0853 *
      11) PARED>=14.5 1364  149348500 496.5171 *
   3) JOYSCIE>=-0.1626 7487  948249800 521.3828
     6) PARED< 14.5 3725  400876300 495.4454
      12) nbooks< 2.5 1049  100022100 462.9746 *
      13) nbooks>=2.5 2676  276643400 506.8371 *
     7) PARED>=14.5 3762  464959900 543.9905
      14) nbooks< 3.5 1480  177490100 516.6616 *
      15) nbooks>=3.5 2282  252445100 561.0575 *
```


100%
yes
JOYSCIE < -0.16
no
464
37%
521
63%
nbooks < 2.5
PARED < 14
477
26%
495
29%
544
34%
PARED < 14
nbooks < 2.5
nbooks < 3.5
432
11%
461
14%
497
12%
463
8%
507
22%
517
13%
561
21%

All the variables



All the variables



Share and share alike