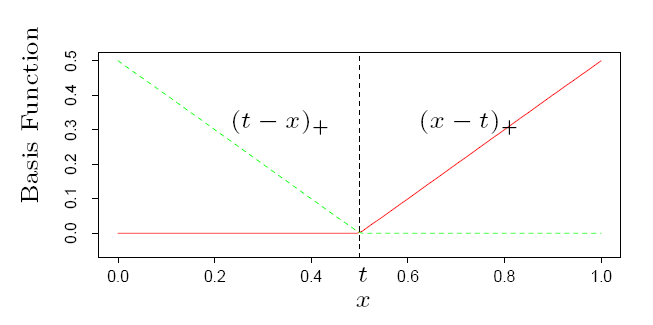
# Multivariate Adaptive Regression Splines (MARS)

# 5.1 – The Multivariate Adaptive Regression Spline (MARS) Model

# In MARS we fit a multivariate regression model of the form,

# where the are terms based on one or two predictors with a very specific form. All the terms are created using a *basis* of *Hockey Stick* functions. Recall that Hockey Stick functions for a single numeric predictor () are given by one of the two piecewise linear function defined below.



# or

# We may also choose to include *interaction* terms in our MARS model. Recall that interaction terms are products of terms in the MLR model. Interaction terms in the MARS model will simply be the product of any two Hockey Stick basis functions where the predictors they are based on differ. For example an interaction terms based on two arbitrary predictors might be given by,

# If there are factors (i.e. dummy variables corresponding to levels of nominal/ordinal predictors) to be used in the MARS model then the model needs to be expanded to include these terms. Thus the terms could include standard dummy terms of the form,

# and as a result if we allow interaction terms in our model, they could have the form

# 

or

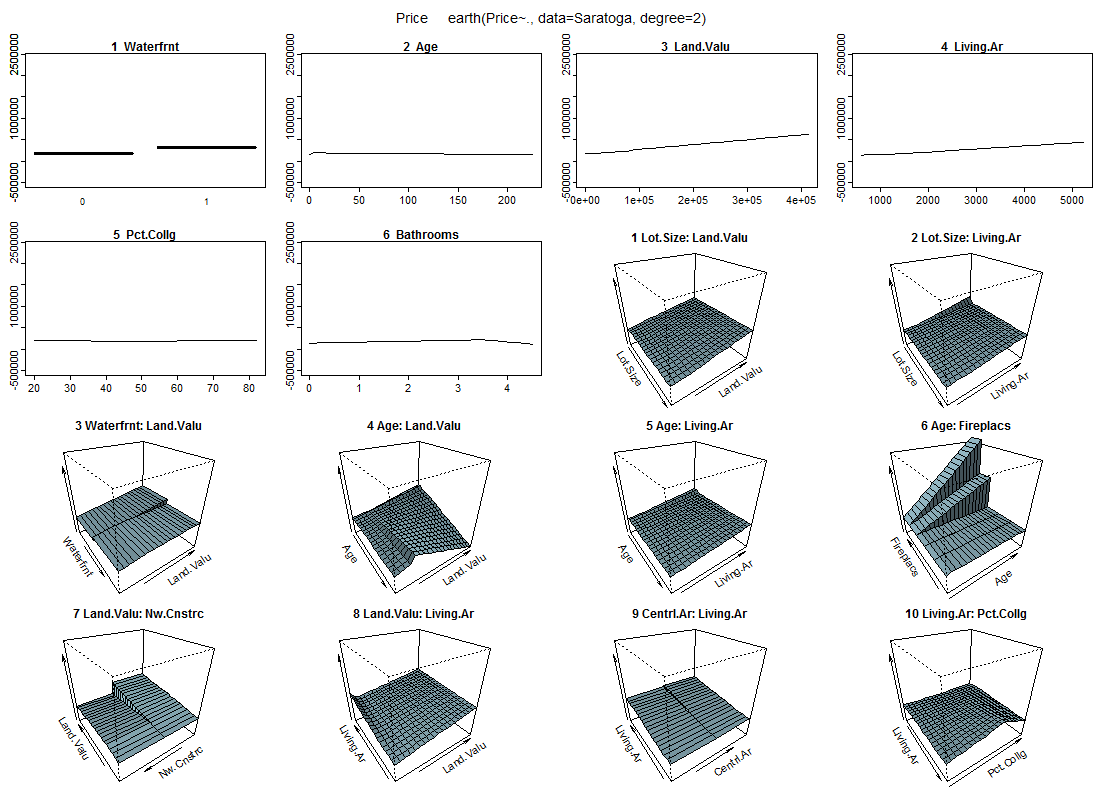
# where,

# Below are plots of the terms created using the MARS algorithm fit the LA Basin ozone data considered in earlier sections of the notes. Notice that the functional form of the predictor day (day of year, 1 – 330) comes from adding several hockey stick functions together to produce a nonlinear function . Also notice it an interaction term based in inversion base height (inbh) and humidity (hum) was also chosen. It appears the fitted surface displaying this interaction, is based upon several products of hockey stick functions for each predictor.

# Example of a MARS fit to the LA Basin Ozone Concentration Data

# Notice the plots are all on the same scale, this allows us to determine visually the most important predictors and their associated terms in the MARS model by considering the range of the vertical axis.

As a second example consider MARS fit the Saratoga, NY home price data. Notice that the appearance of the terms based on the predictor waterfront (1 = home is on a waterfront, 0 = home is not on waterfront) and the interaction terms between factors and numeric variables (e.g. waterfront and land value & land value and new construction) by the MARS model.

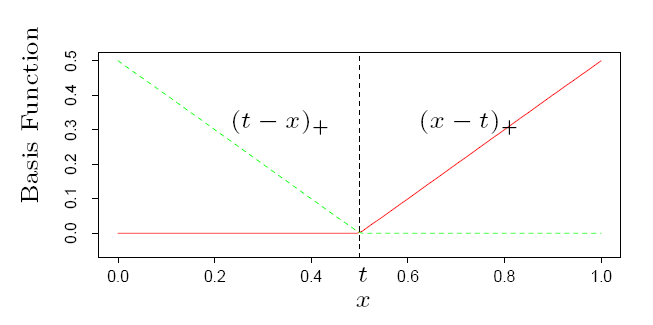


**5.2 – The MARS Algorithm**

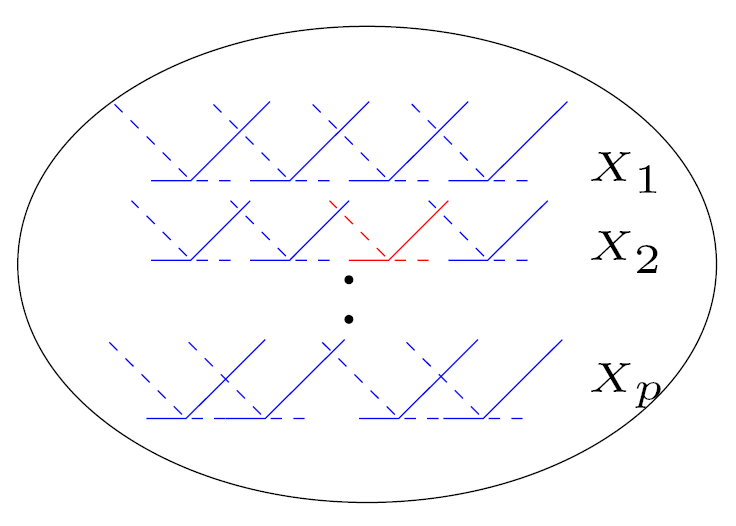
To simplify the initial discussion of how MARS arrives at a “final” model we will first consider the case where all predictors are numeric variables. The basic idea is to use a “greedy” forward search algorithm to select the final terms in the MARS model with a given number of non-constant (i.e. non-intercept) terms (). Greedy algorithms choose the best possible course of action at each stage. In model development this would mean adding the best possible term to the model at each stage. Forward selection means that we start with the intercept only model, i.e. , and add the best term from the candidate pool first and add successive terms to the model in a greedy fashion. At no point does the algorithm consider dropping a term once it is added, thus it is “greedy”.

What are the candidate terms in the MARS algorithm? In the case of all numeric predictors with no replicate values (i.e. each predictor consists of unique values) the maximum number of basis terms is .

**Why?**

Each Hockey Stick function in the basis has the form shown below and is determined by the choice of predictor () and the ***knot*** value (). The knot choice () seems overwhelming, as for any predictor there is seemingly an infinitely many choices for the values for within the observed range of a given predictor . However, if each predictor has unique values and we choose knots equal to observed predictor values only, then we have only choices of for each predictor.  


Below is a picture of the non-constant basis functions when we have numeric predictors with unique values each. As stated above we will have at most functions total in our basis.



If there are nominal/ordinal predictors in the pool then the dummy variables that define their levels would constitute additional terms to be added to model. These terms could just be the factor levels themselves or interactions with other terms.

Terms are chosen from set of basis functions shown in diagram above in pairs, i.e. terms will consist of terms created from both .

For example, suppose the first terms added are then at the next stage a terms of a pair of products of the form and would be added where would have one of the following forms:

Essentially the next term added will either be another pair of basis functions based on predictor (), interactions terms based on another predictor () with the term , or interaction terms based on () formed with . This process repeats until either the maximum number of terms () is reached or the GCV criterion fails to improve when a new terms is added. The GCV criterion penalizes models with too many terms.

Here = equals a model complexity measure which is the number of ***effective*** parameters in the model. Aside from the usual we also incur a penalty for the choice of the knot values (). When we have interaction terms in the model each unique knot choice is equivalent to estimating *three* parameters () and when we only include linear terms (i.e. non-interaction terms) then each knot choice is equivalent to estimating *two* () parameters. If there are a total non-constant terms as well as knots () chosen then . Rival models are compared on the basis of criterion, smaller is better.

Usually we choose the number of terms to include in our final model, but this choice is rather arbitrary. To improve the MARS model selection process, we typically also specify a minimum model size and use backward selection procedures to choose good candidate submodels once the forward search process starts. In the end will have several “good” candidates with number of terms ranging from the specified minimum model size to the maximum model size, and we can then choose the one that has the smallest GCV score.

**Summary of MARS Algorithm**

**Step 0** - .

**Step 1** – **Forward Selection Stage (repeat)**

* Decide which basis functions are candidates at each stage.
* Decide which basis functions can added to existing model.
* Add best candidate terms (in pairs).
* Fit model and compute
* If model is better in terms score, save it and continue
* Stop if the model is not better in terms of score or if maximum model size is reached.

**Step 2 – Backward Selection Stage (repeat)**

* Decide which basis functions can be removed from the model.
* Remove the one that is deemed the worst.
* Fit the submodel with worst term delete and compute .
* If the model is better than the one selected via forward selection, save it.
* Continue to drop terms until minimum model size is reached, saving the best ones along the way.

We can still use cross-validation methods discussed in Section 3 to help choose amongst the models returns from the MARS algorithm described above as it will likely produce several candidate models that seem “good”.

Rather than minimize the criterion we can attempt to maximize the Generalized R-square () which is given by,

**5.3 – MARS Examples**

We now consider examples of using MARS to develop predictive models for a numeric response. In these examples we will be using functions contained in the packages earth and plotmo, thus you will need to install and load these packages to go through these examples yourself. We also look at some of the many options that can specified when fitting MARS models using the earth() command.

> library(earth)

Loading required package: plotmo 🡨 the earth package automatically requires plotmo.  
Loading required package: plotrix

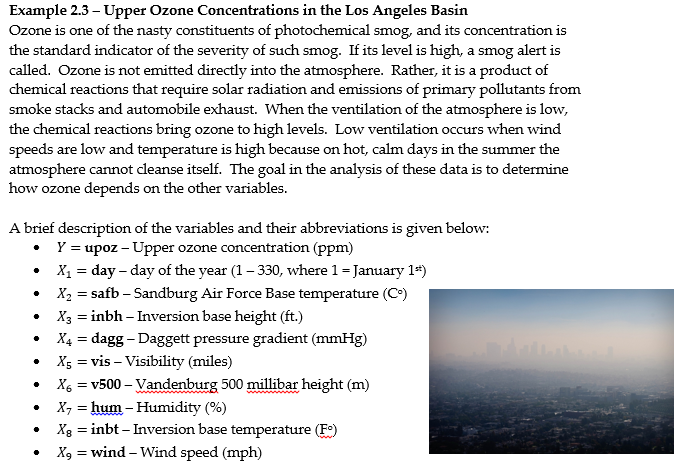
Loading required package: TeachingDemos

Warning message: package ‘earth’ was built under R version 3.2.3

**Example 5.1 – LA Basin Ozone Data**

> Ozone = read.table(file.choose(),header=T,sep=”,”)  
  
> names(Ozdata)

[1] "upoz" "day" "v500" "wind" "hum" "safb" "inbh" "dagg" "inbt" "vis"



> head(Ozdata)

upoz day v500 wind hum safb inbh dagg inbt vis

1 3 3 5710 4 28 40 2693 -25 87 250

2 5 4 5700 3 37 45 590 -24 128 100

3 5 5 5760 3 51 54 1450 25 139 60

4 6 6 5720 4 69 35 1568 15 121 60

5 4 7 5790 6 19 45 2631 -33 123 100

6 4 8 5790 3 25 55 554 -28 182 250

> str(Ozdata)

'data.frame': 330 obs. of 10 variables:

$ day : num 3 4 5 6 7 8 9 10 11 12 ...

$ v500: num 5710 5700 5760 5720 5790 5790 5700 5700 5770 5720 ...

$ wind: num 4 3 3 4 6 3 3 3 8 3 ...

$ hum : num 28 37 51 69 19 25 73 59 27 44 ...

$ safb: num 40 45 54 35 45 55 41 44 54 51 ...

$ inbh: num 2693 590 1450 1568 2631 ...

$ dagg: num -25 -24 25 15 -33 -28 23 -2 -19 9 ...

$ inbt: num 87 128 139 121 123 182 114 91 92 173 ...

$ vis : num 250 100 60 60 100 250 120 120 120 150 ...

$ upoz: num 3 5 5 6 4 4 6 7 4 6 ...

From the help file for the earth() command:  
earth(formula, data = NULL,

weights = NULL, wp = NULL, subset = NULL,

na.action = na.fail,

pmethod = c("backward", "none", "exhaustive", "forward", "seqrep", "cv"),

keepxy = FALSE, trace = 0, glm = NULL, degree = 1, nprune = NULL,

ncross=1, nfold=0, stratify=TRUE,

varmod.method = "none", varmod.exponent = 1,

varmod.conv = 1, varmod.clamp = .1, varmod.minspan = -3,

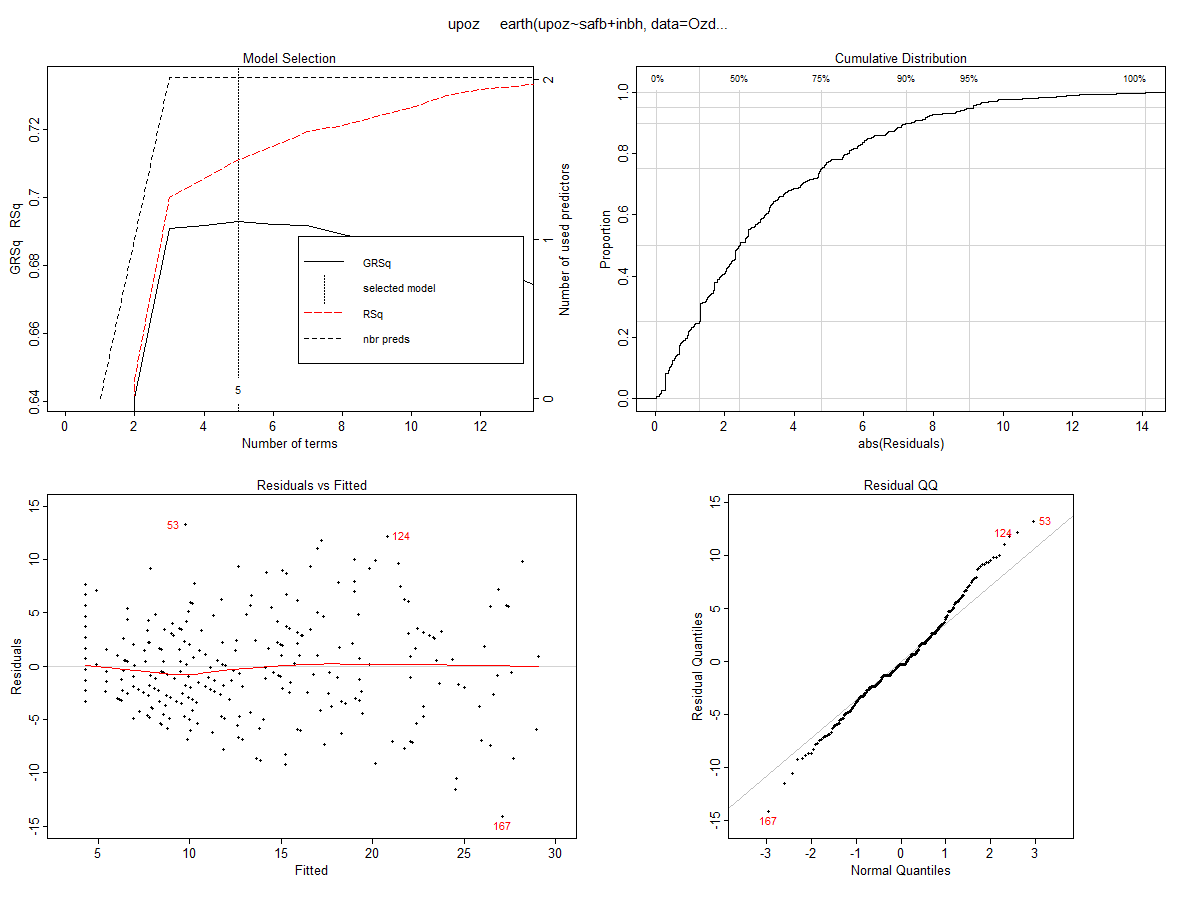
Scale.y = (NCOL(y)==1), ...)

The basic call, assuming all of the other variables aside from the response are candidate predictors, is

> earth(y~.,data=*dataframe*)

To better understand how the MARS algorithm builds terms and what the final functional form might look like consider a model for using and as the only two predictors.

> oz.mars = earth(upoz~safb+inbh,degree=2,data=Ozdata)

> plot(oz.mars)  


> summary(oz.mars)

Call: earth(formula=upoz~safb+inbh, data=Ozdata2, degree=2)

coefficients

(Intercept) 10.3358863

h(safb-58) 0.5767490

h(1046-inbh) -0.0036303

h(inbh-1046) -0.0015256

h(safb-72) \* h(inbh-1046) -0.0003062

Selected 5 of 18 terms, and 2 of 2 predictors

Termination condition: Reached nk 21

Importance: safb, inbh

Number of terms at each degree of interaction: 1 3 1

GCV 19.76711 RSS 6095.519 GRSq 0.6929411 RSq 0.7113236

**Plotting the fitted surface for a simple MARS model to L.A. Basin Ozone data**

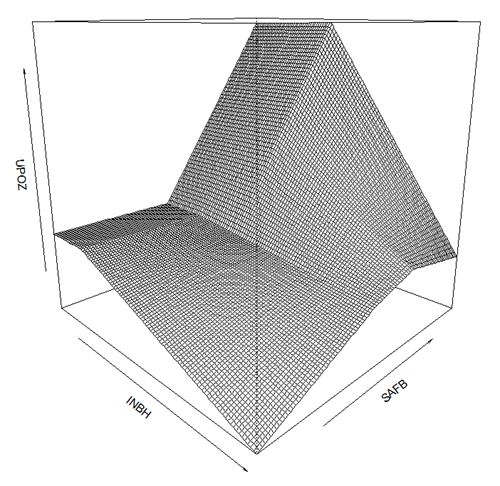
> x1 = seq(min(inbh),max(inbh),length=100)

> x2 = seq(min(safb),max(safb),length=100)

> x = expand.grid(inbh=x1,safb=x2)

> ypred = predict(oz.mars,newdata=x)

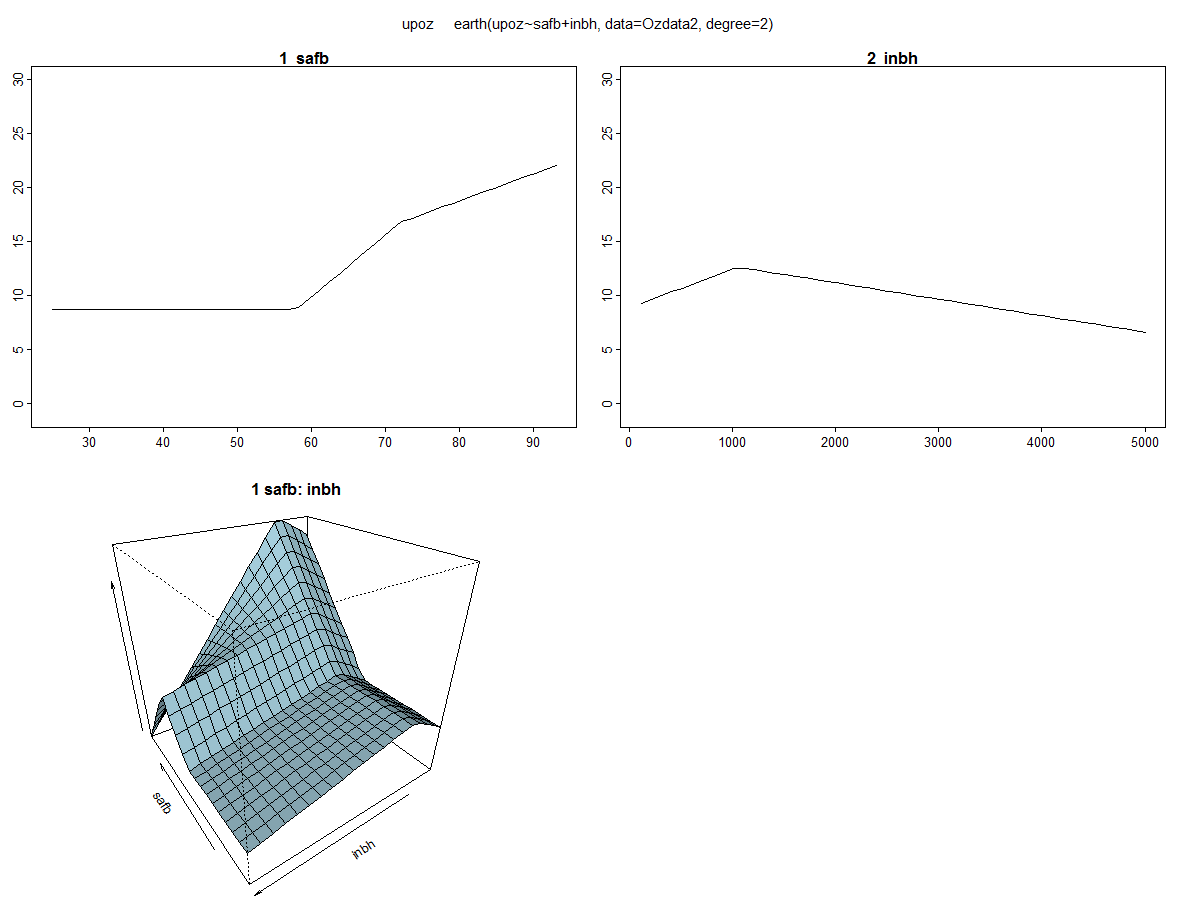
> persp(x1,x2,z=matrix(ypred,100,100),theta=45,xlab="INBH",

+ ylab="SAFB",zlab="UPOZ")  


> plotmo(oz.mars)

grid: safb inbh

62 2112.5

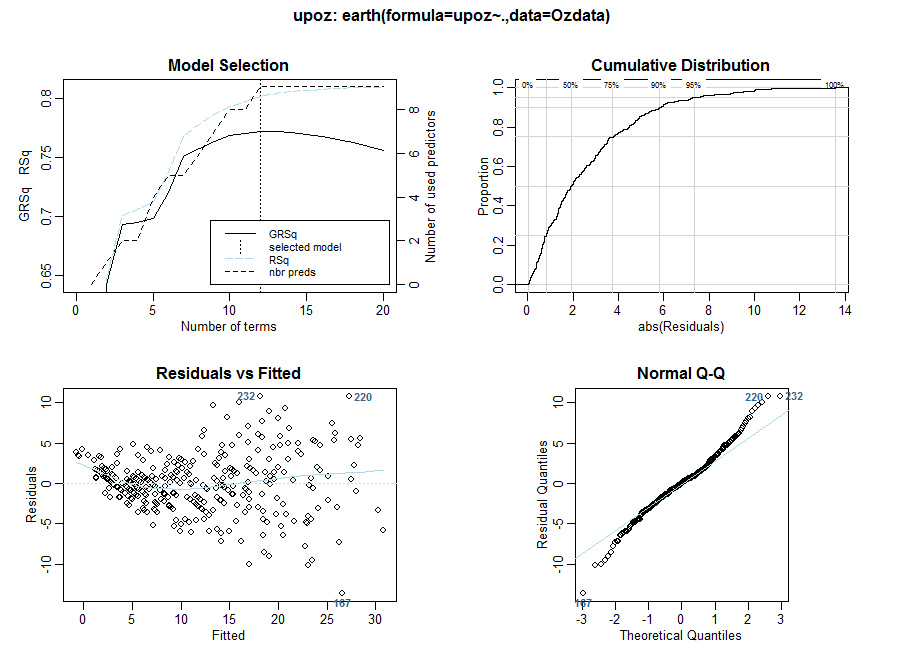


Next we will fit a MARS model using all candidate predictors and examine model development issues in more detail.

The earth()will fit a MARS regression model using similar nomenclature to that used to fit OLS models using the lm() function.

> oz.mars = earth(upoz~.,data=Ozdata) 🡨 basic MARS call, nothing fancy

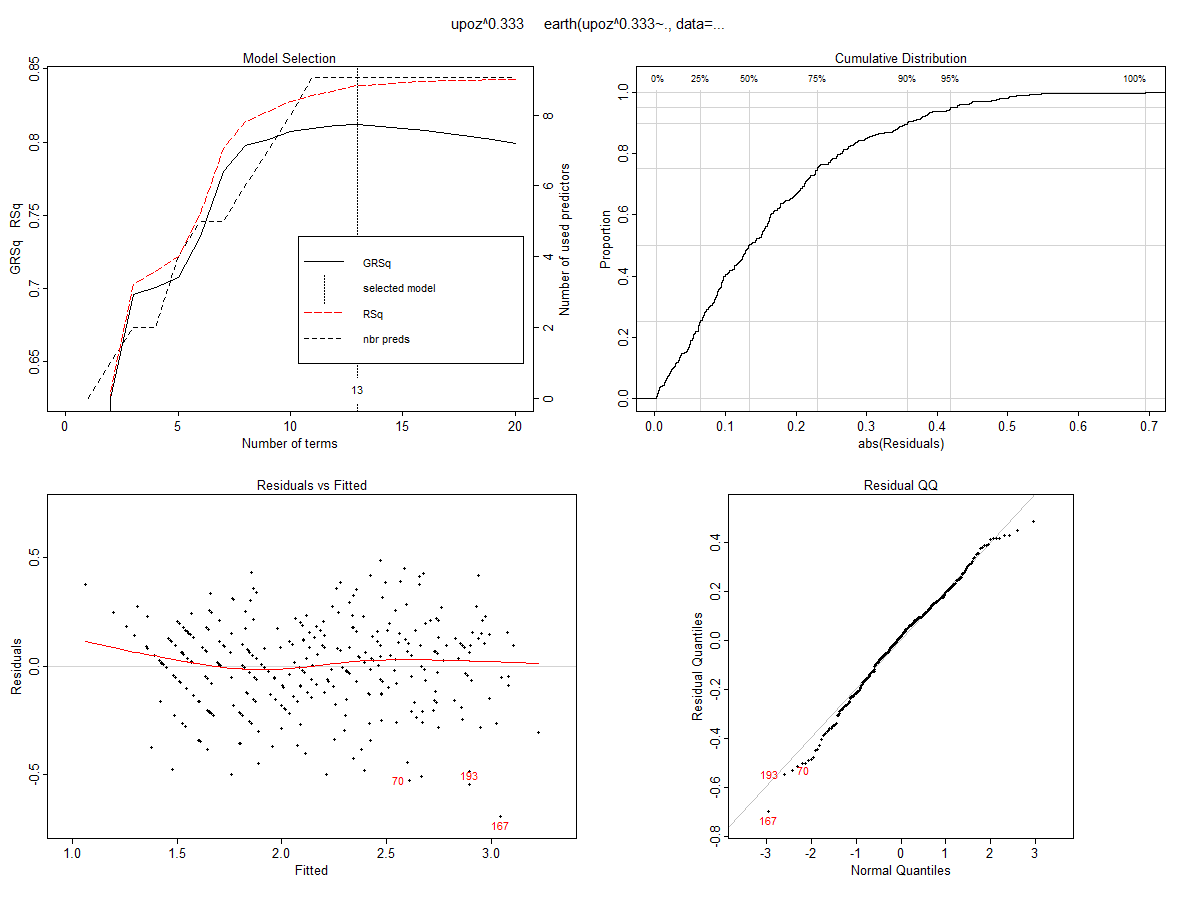
> plot(oz.mars)



We still need to consider response transformations if needed. Here the heteroscedasticity is obvious, along with some degree of curvature. We will transform response using a cube root transformation and rerun the basic MARS fit.

> oz.mars2 = earth(upoz^.333~.,data=Ozdata)

> plot(oz.mars2)



> summary(oz.mars2)

Call: earth(formula=upoz^0.333~., data=Ozdata)

coefficients

(Intercept) 2.45740714

h(59-day) -0.00915281

h(day-59) 0.00274801

h(day-115) -0.00549099

h(5770-v500) -0.00089419

h(9-wind) 0.01989047

h(54-hum) -0.00566332

h(safb-54) 0.01548911

h(1105-inbh) -0.00021818

h(10-dagg) -0.00331433

h(dagg-10) -0.00673141

h(256-inbt) -0.00222275

h(200-vis) 0.00126372

Selected 13 of 20 terms, and 9 of 9 predictors

Termination condition: Reached nk 21

Importance: safb, day, v500, hum, dagg, inbh, vis, inbt, wind

Number of terms at each degree of interaction: 1 12 (additive model)

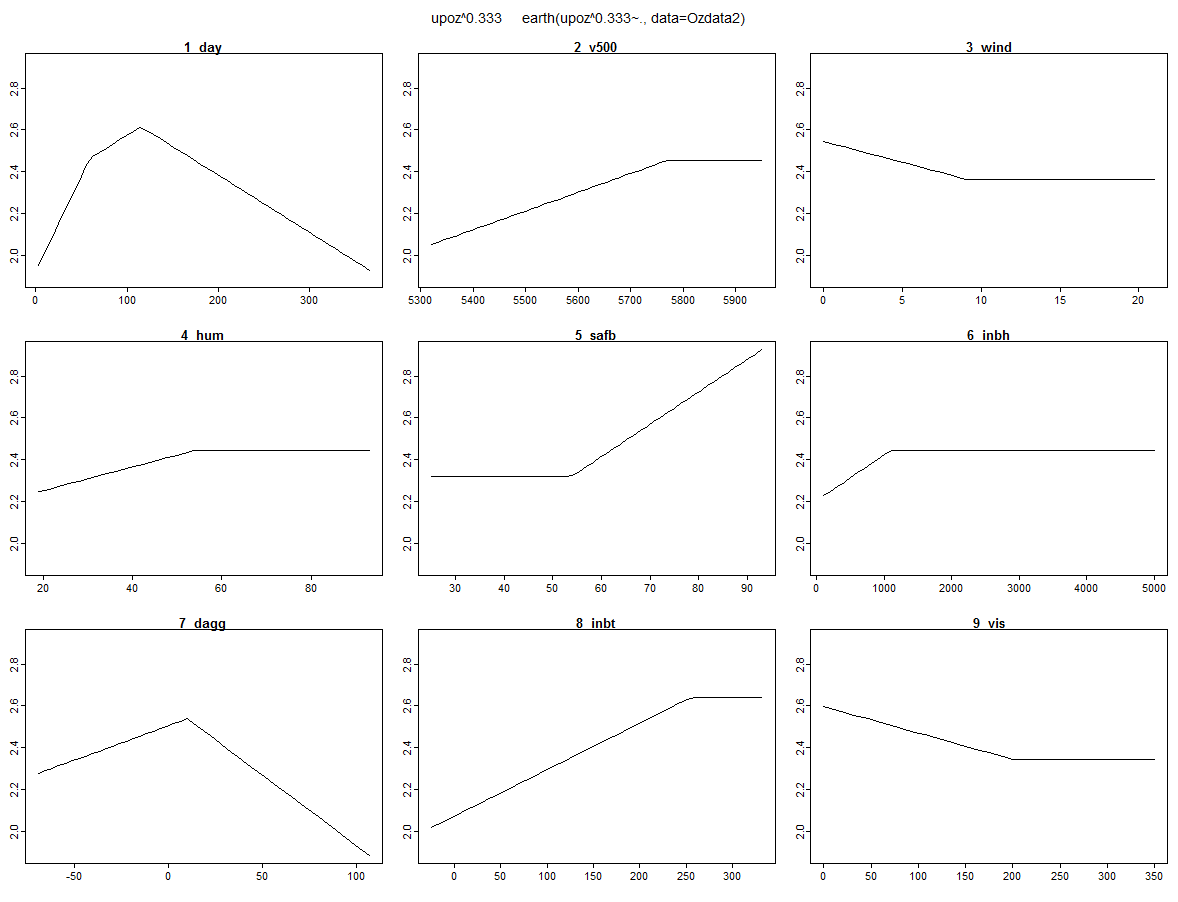
GCV 0.05088398 RSS 14.34389 GRSq 0.8119464 RSq 0.8383821

To see what the terms look like we can use the function plotmo()from the plotmo package. The command plotmo()stands for **plot** **mo**del which can be used with a variety of model types in R. See the plotmo help page for more details on which model types it can plot.

> plotmo(oz.mars2)

grid: day v500 wind hum safb inbh dagg inbt vis

177.5 5760 5 64 62 2112.5 24 167.5 120



The equation below gives the following fitted model:

> summary(oz.mars2)

Call: earth(formula=upoz^0.333~., data=Ozdata)

coefficients

(Intercept) 2.45740714

h(59-day) -0.00915281

h(day-59) 0.00274801

h(day-115) -0.00549099

h(5770-v500) -0.00089419

h(9-wind) 0.01989047

h(54-hum) -0.00566332

h(safb-54) 0.01548911

h(1105-inbh) -0.00021818

h(10-dagg) -0.00331433

h(dagg-10) -0.00673141

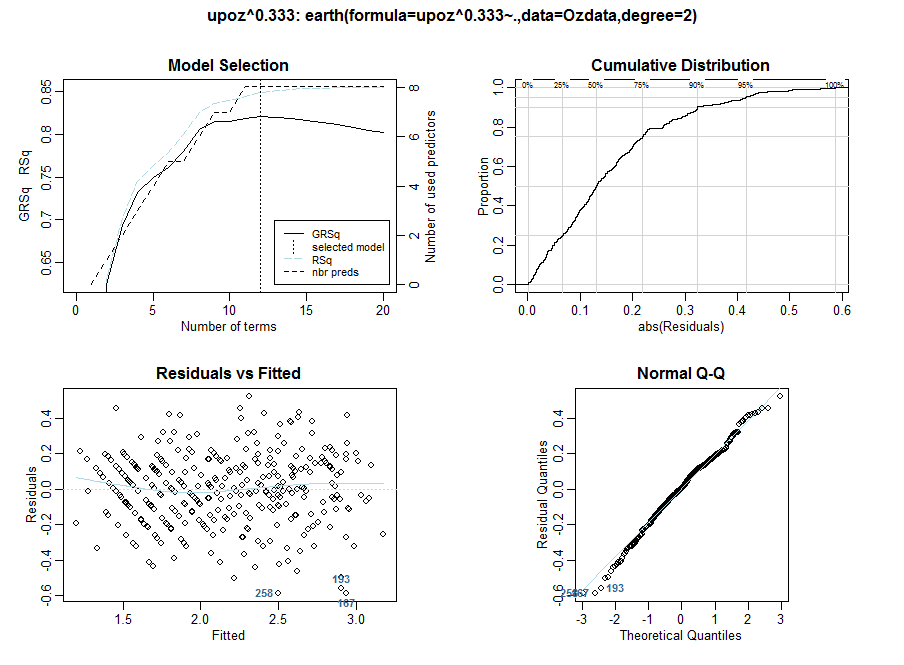
h(256-inbt) -0.00222275

h(200-vis) 0.00126372

We now consider adding higher order terms, i.e. potential interactions between the covariates possibly of the form .

> oz.mars3 = earth(upoz^.333~.,degree=2,data=Ozdata)

> plot(oz.mars3)



> summary(oz.mars3)

Call: earth(formula=upoz^0.333~., data=Ozdata2, degree=2)

coefficients

(Intercept) 2.53284967

h(59-day) -0.00804964

h(day-59) 0.00357229

h(day-103) -0.00580143

h(5770-v500) -0.00113987

h(67-hum) -0.00658098

h(safb-54) 0.01531530

h(1105-inbh) -0.00018983

h(10-dagg) -0.00330286

h(dagg-10) -0.00589822

h(247-inbt) -0.00152413

h(200-vis) 0.00111110

h(hum-50) \* h(inbh-1105) -0.00000330

Selected 13 of 20 terms, and 8 of 9 predictors

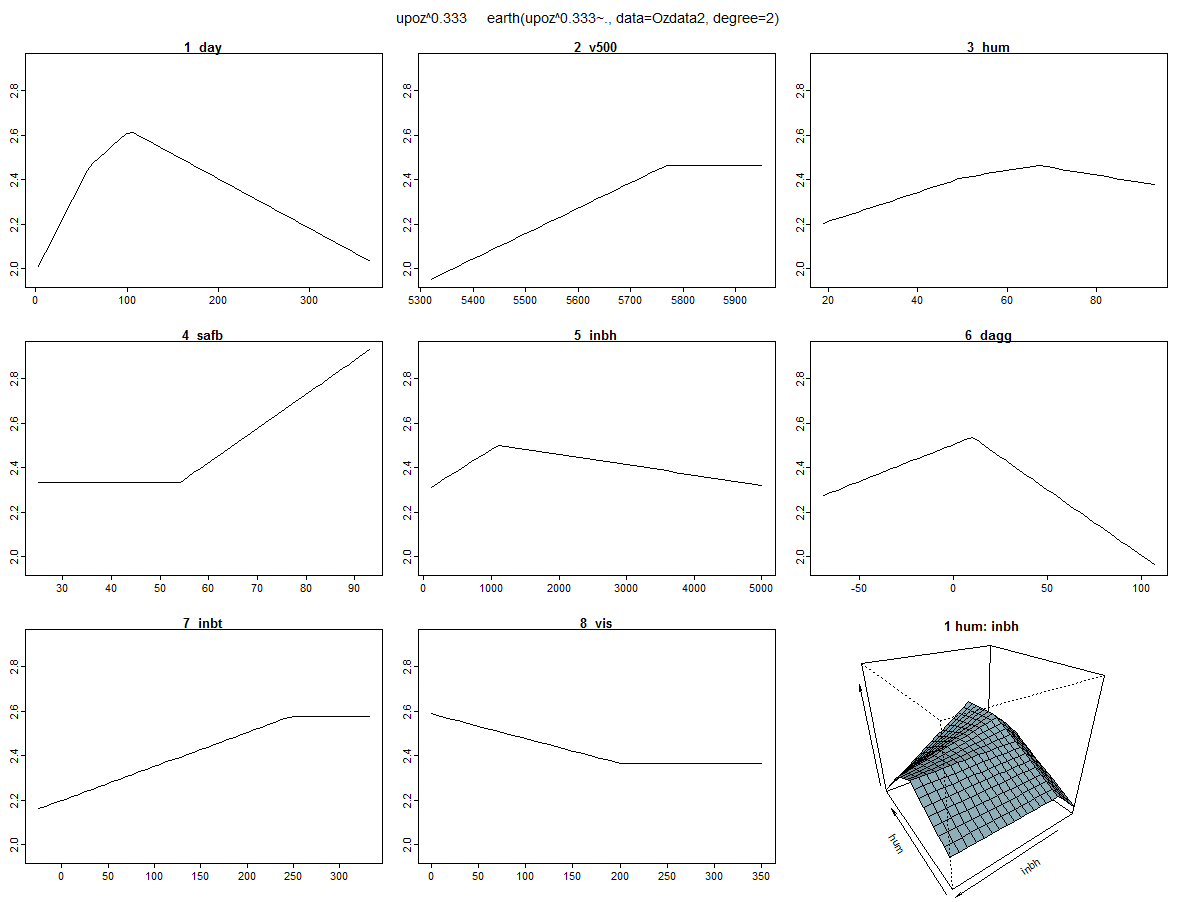
Termination condition: Reached nk 21

Importance: safb, inbh, hum, v500, day, dagg, vis, inbt, wind-unused

Number of terms at each degree of interaction: 1 11 1

GCV 0.04849227 RSS 13.13714 GRSq 0.8207855 RSq 0.8519789

> plotmo(oz.mars3)



Here we can see the interaction term between humidity and inversion base height plotted. It is also interesting to note that this model does not include a term based on wind speed.

There is also a slight improvement in generalized R-square from the first degree term only model. As with any flexible modeling strategy cross-validation is critical. The earth function has some built-in functionality for performing cross-validation, which we will consider below.

We can force MARS to consider larger models by setting the option nk = max. terms condition. The previous two fits to these data stopped the forward selection at nk = 21. We can increase to force the algorithm to consider larger models during the forward selection phase.

> summary(oz.mars4)

Call: earth(formula=upoz^0.333~., data=Ozdata, degree=2, nk=30)

coefficients

(Intercept) 2.46914853

h(59-day) -0.00790062

h(day-59) 0.00345259

h(day-103) -0.00544060

h(5770-v500) -0.00104231

h(67-hum) -0.00659119

h(hum-67) 0.00883007

h(safb-54) 0.01387131

h(10-dagg) -0.00285264

h(dagg-10) -0.00519803

h(247-inbt) -0.00135723

h(200-vis) 0.00096317

h(day-240) \* h(1105-inbh) -0.00000348

h(wind-7) \* h(200-vis) -0.00078953

h(hum-67) \* h(73-safb) -0.00076507

h(hum-50) \* h(inbh-1105) -0.00000265

h(inbh-1105) \* h(40-vis) 0.00001645

Selected 17 of 28 terms, and 9 of 9 predictors

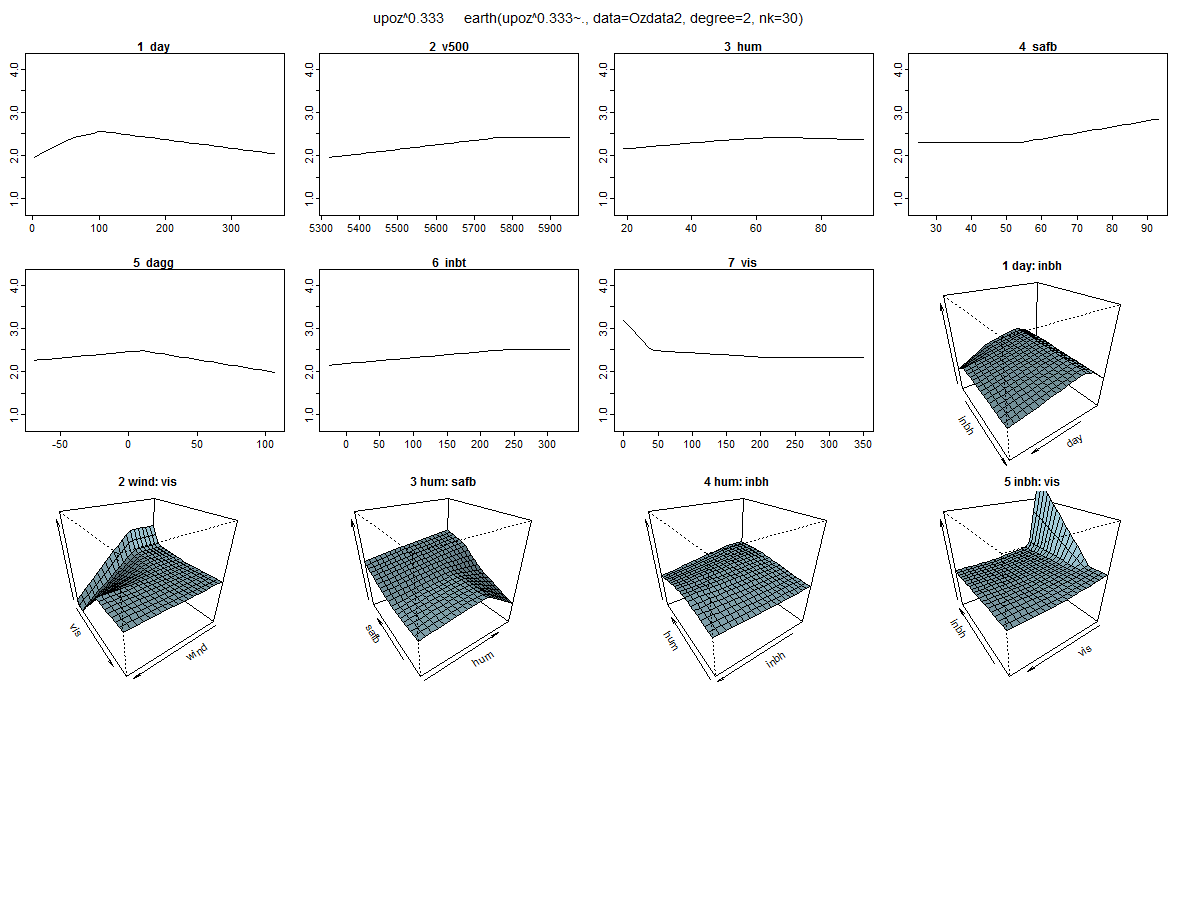
Termination condition: Reached nk 30

Importance: safb, inbh, hum, v500, day, dagg, vis, wind, inbt

Number of terms at each degree of interaction: 1 11 5

GCV 0.04702638 RSS 11.90209 GRSq 0.8262031 RSq 0.8658947

> plotmo(oz.mars4)



**Built-in Cross-validation with MARS using the earth package**

The earth function has built in cross-validation functionality that can be plotted and used to choose an optimal number of terms for a MARS model. This handout will demonstrate their use and some additional optional arguments that are useful in the model building process.

We will use the L.A. Basin ozone concentration data again in our examples.

> oz.mars = earth(upoz^0.333~.,degree=2,nk=30,nfold=10,data=Ozdata)

> summary(oz.mars)

Call: earth(formula=upoz^0.333~., data=Ozdata, degree=2, nfold=10, nk=30)

coefficients

(Intercept) 2.46914853

h(59-day) -0.00790062

h(day-59) 0.00345259

h(day-103) -0.00544060

h(5770-v500) -0.00104231

h(67-hum) -0.00659119

h(hum-67) 0.00883007

h(safb-54) 0.01387131

h(10-dagg) -0.00285264

h(dagg-10) -0.00519803

h(247-inbt) -0.00135723

h(200-vis) 0.00096317

h(day-240) \* h(1105-inbh) -0.00000348

h(wind-7) \* h(200-vis) -0.00078953

h(hum-67) \* h(73-safb) -0.00076507

h(hum-50) \* h(inbh-1105) -0.00000265

h(inbh-1105) \* h(40-vis) 0.00001645

Selected 17 of 28 terms, and 9 of 9 predictors

Termination condition: Reached nk 30

Importance: safb, inbh, hum, v500, day, dagg, vis, wind, inbt

Number of terms at each degree of interaction: 1 11 5

GCV 0.04702638 RSS 11.90209 GRSq 0.8262031 RSq 0.8658947 CVRSq 0.7810595

Note: the cross-validation sd's below are standard deviations across folds

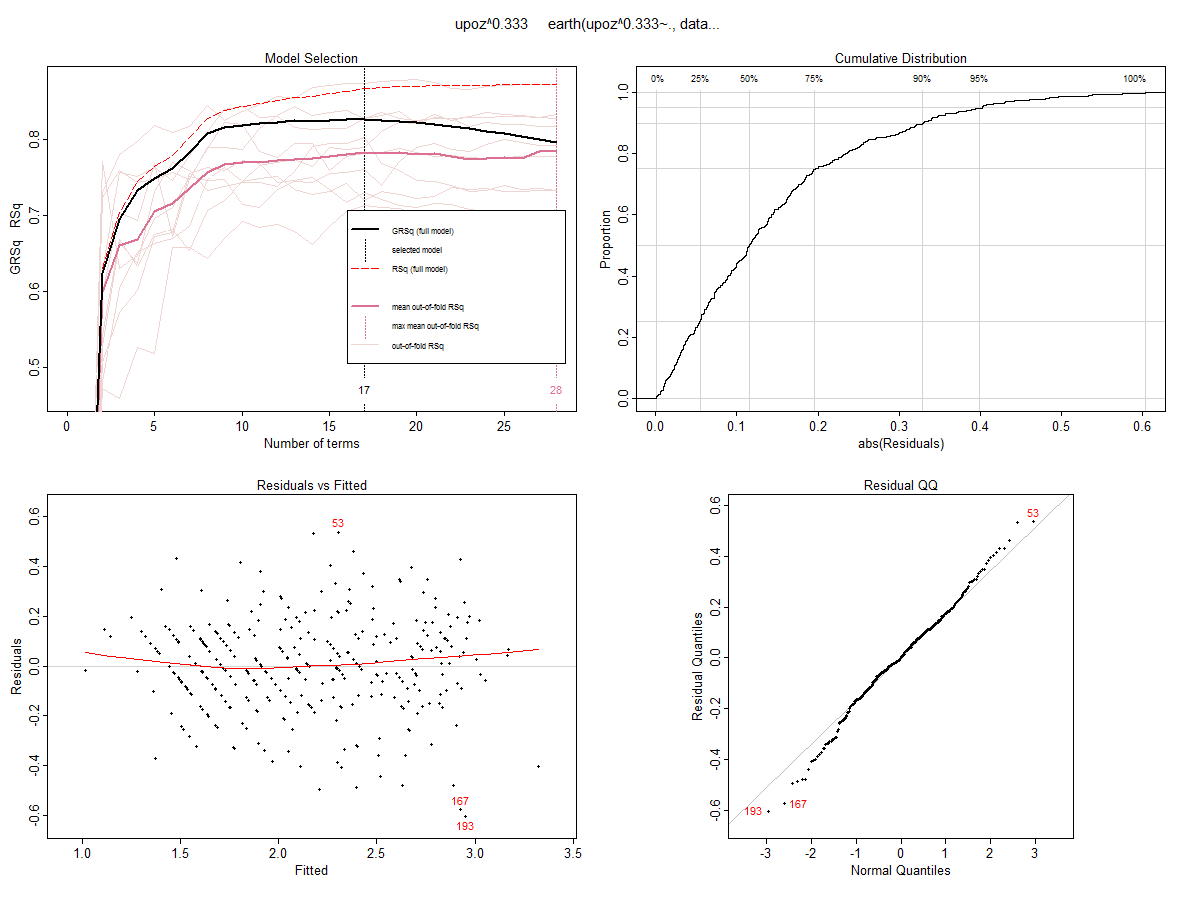
Cross validation: nterms 15.90 sd 0.99 nvars 8.60 sd 0.70

CVRSq sd MaxErr sd

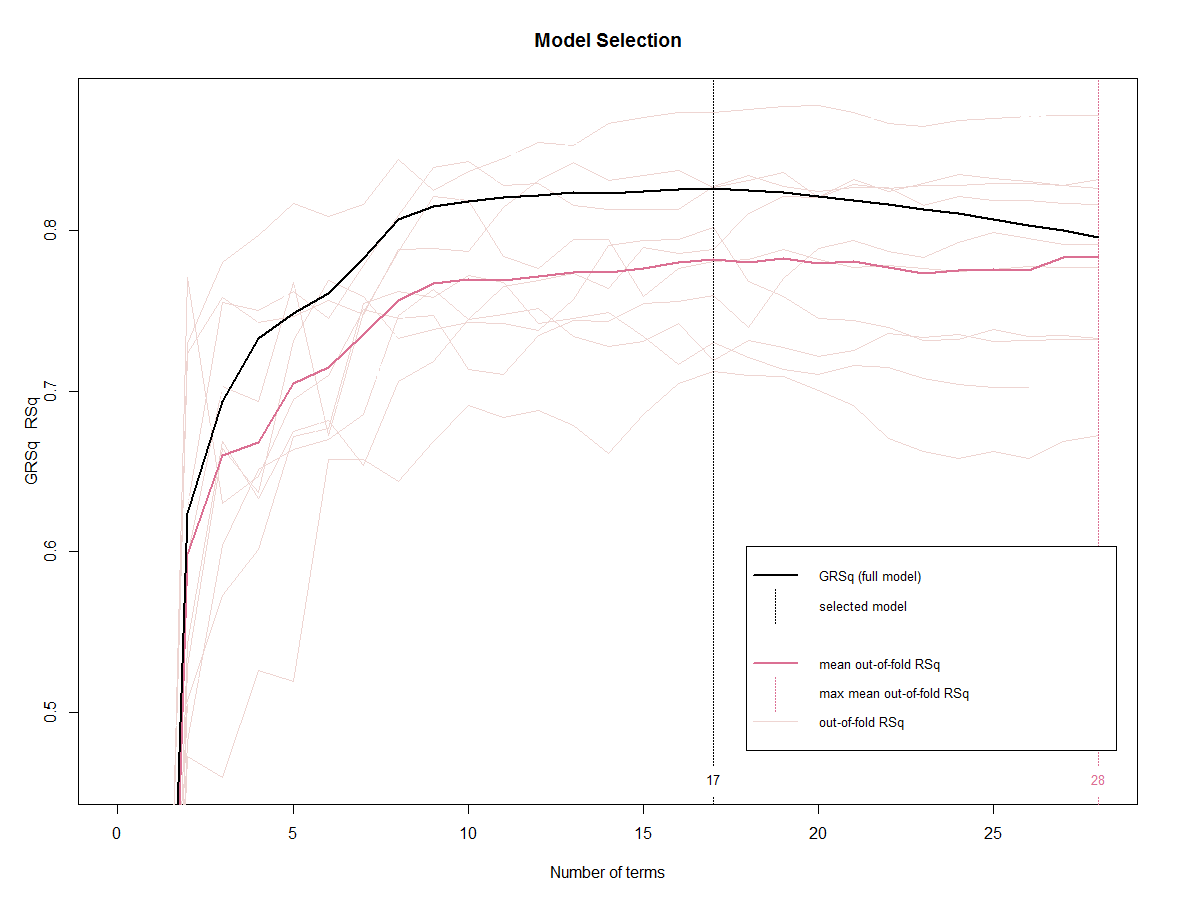
0.781 0.041 -0.673 0.399

To examine cross-validation results graphically it is best to save the results using the option keepxy=T option when using nfold to perform the cross-validation.

> oz.mars = earth(upoz^.333~.,data=Ozdata,degree=2,nfold=10,keepxy=T)



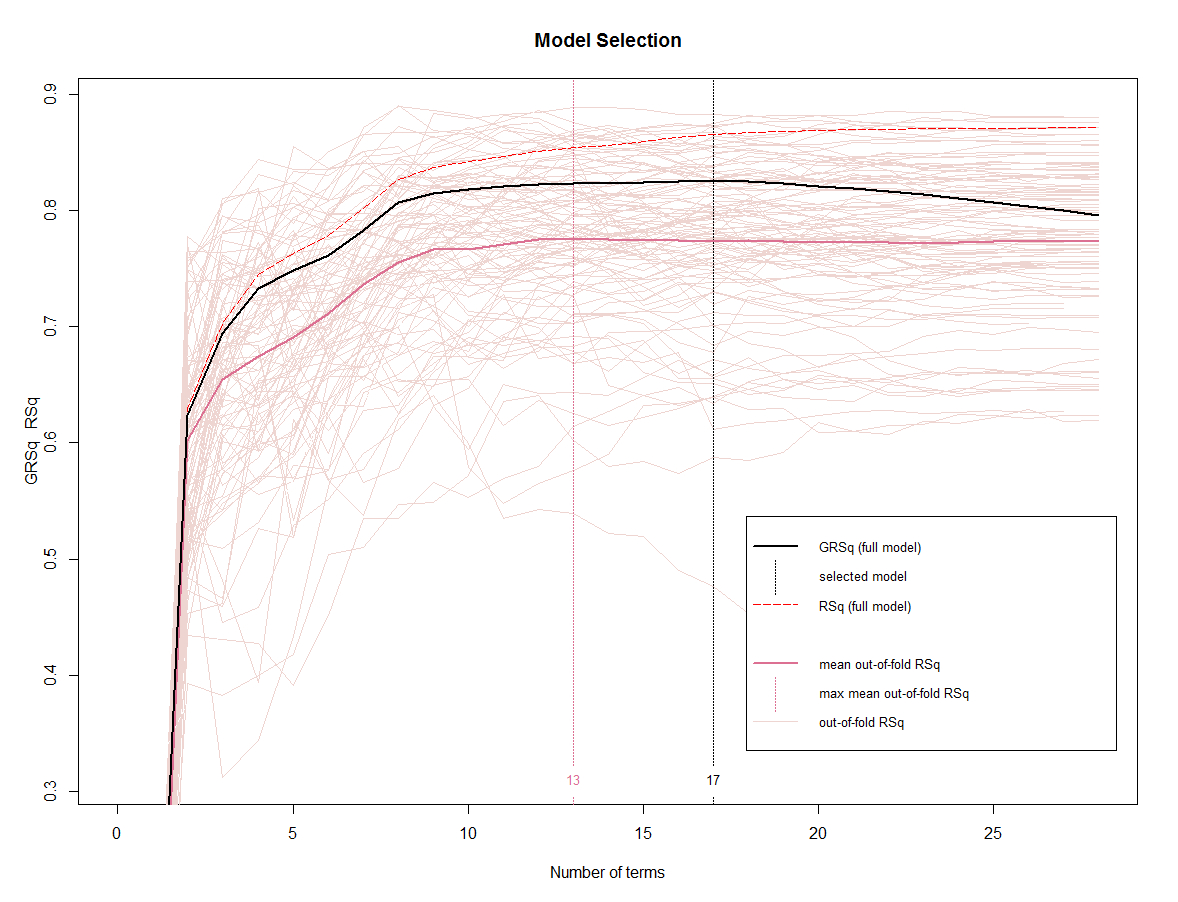
> plot(oz.mars,which=1,col.rsq=0) 🡨 displays only the model selection plot. The col.rsq = 0   
 option will block out the R2 portion of the plot.



We can choose a different value for *k* in the *k-fold* cross-validation performed by the nfold option. We can also use the ncross option to perform the *k-fold* cross-validation ncross times. This will allow us to see the degree of variability in the *k-fold* cross-validation results.

> oz.mars = earth(upoz^.333~.,data=Ozdata,degree=2,ncross=10,nfold=5,keepxy=T)

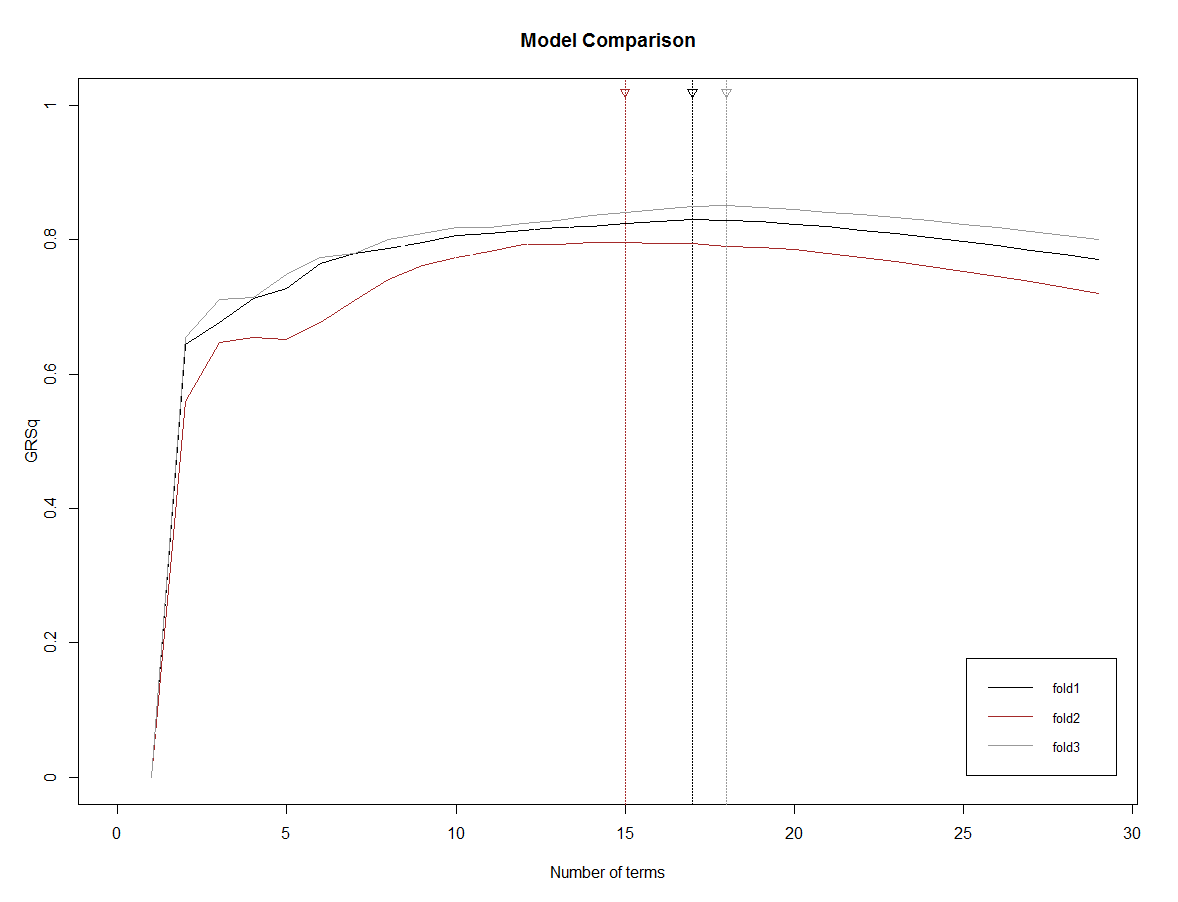
> plot(oz.mars,which=1)



When using a small value for k-fold cross-validation we can plot the results of each fold individually as follows:

> oz.mars = earth(upoz^.333~.,data=Ozdata,degree=2,nfold=3,keepxy=T)

> plot.earth.models(oz.mars$cv.list,which=1)

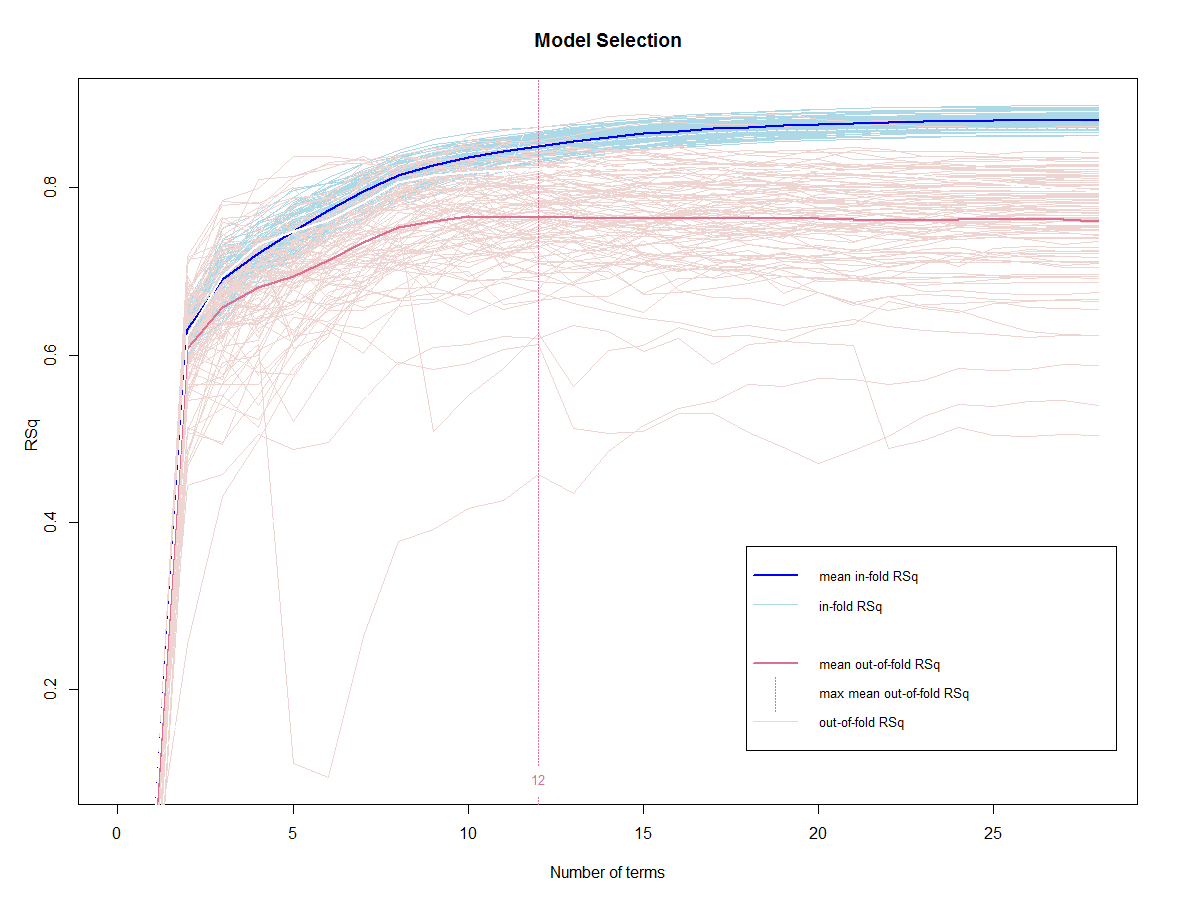


The number of terms chosen varies from fold-to-fold between 15-18 terms it looks like.

> oz.mars = earth(upoz^.333~.,data=Ozdata,degree=2,nk=30,ncross=20,nfold=5,keepxy=T)]

> plot(oz.mars,which=1,col.mean.infold.rsq="blue",col.infold.rsq="lightblue",

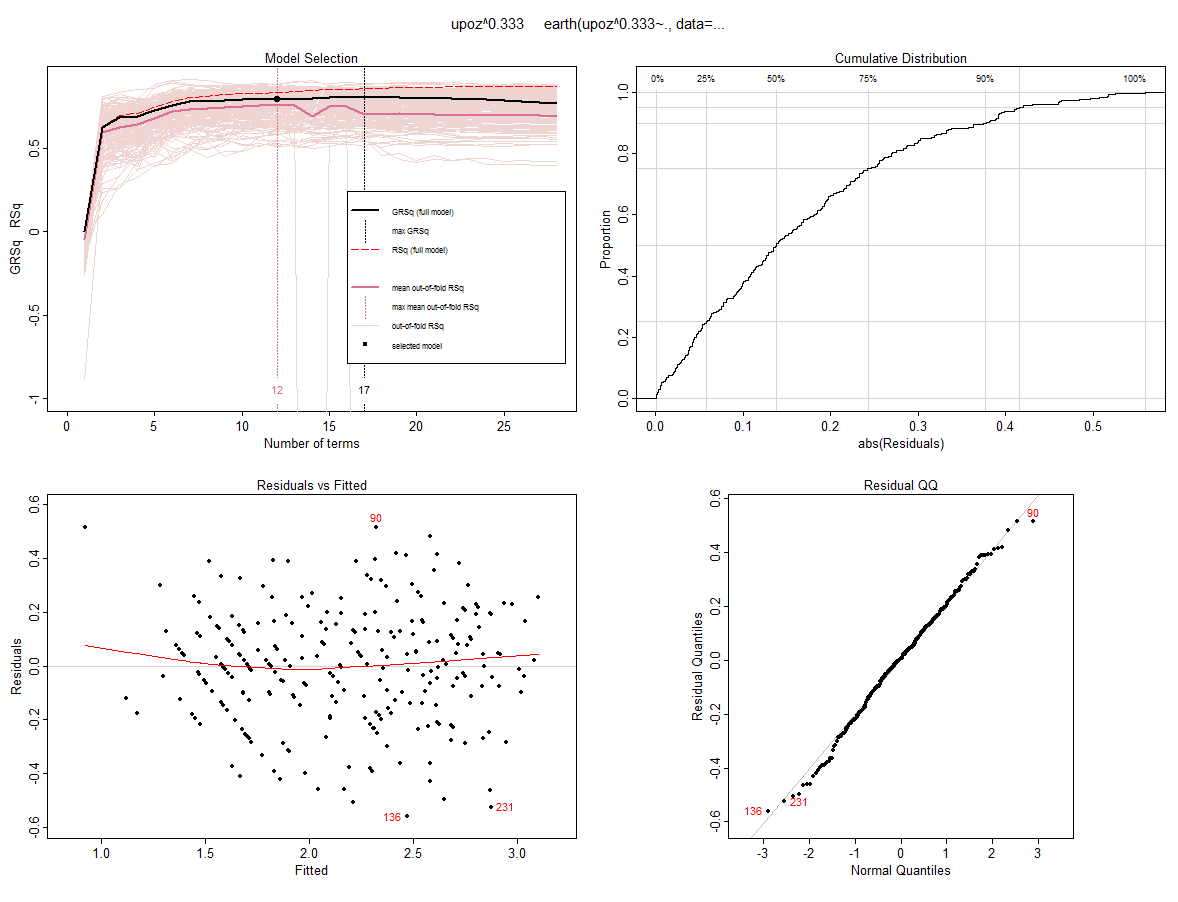
col.grsq=0,col.rsq=0)

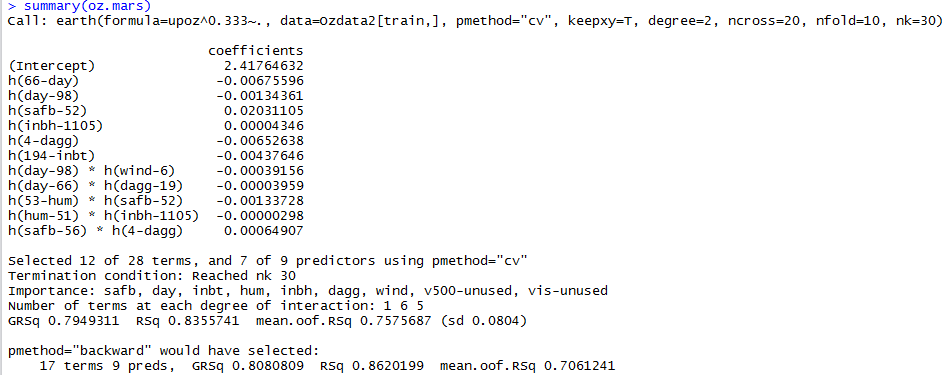


Next we use split-sample cross-validation to compare rival models for the ozone concentrations in the population. As there was some ambiguity about what transformation would be optimal for the response we will consider several, using 20 replicates of 10-fold cross-validation to choose the “optimal” MARS model for each response form.

> train = sample(1:330,size=floor(.80\*330),replace=F)

> oz.mars = earth(upoz^0.333~.,degree=2,nfold=10,data=Ozdata[train,],keepxy=T,nk=30,  
pmethod="cv",ncross=20)





> y.crpred = predict(oz.mars,newdata=Ozdata2[-train,])

> ypred = y.crpred^3

> PredAcc(Ozdata2$upoz[-train],ypred)

> PredAcc(Ozdata2$upoz[-train],ypred)

RMSEP

===============

3.11905

MAE

===============

2.309576

MAPE

===============

27.54405

RMSEP MAE MAPE

1 3.11905 2.309576 27.54405

> PredAcc(Ozdata2$upoz[-train],ypred)

RMSEP

===============

3.361134

MAE

===============

2.513904

MAPE

===============

28.1712

RMSEP MAE MAPE

1 3.361134 2.513904 28.1712

RMSEP

===============

3.285727

MAE

===============

2.411113

MAPE

===============

27.43112

RMSEP MAE MAPE

1 3.285727 2.411113 27.43112

> PredAcc(Ozdata2$upoz[-train],ypred)

RMSEP

===============

3.17807

MAE

===============

2.405936

MAPE

===============

28.34256

RMSEP MAE MAPE

1 3.17807 2.405936 28.34256

The results using square root, fifth root, and log transformations are presented as well. The same strategy was used to find the “optimal” model for each response transformation choice.

**Measuring Variable Importance** – evimp()command

The command evimp function uses three criteria for estimating variable importance in a MARS model. The nsubsets criterion counts the number of model subsets that include the variable, there is one subset for each model size from 1 to selected model size. The rss criterion first calculates the decrease in the RSS for each subset relative to the previous subset. Then for each variable it sums these decreases over all subsets that include the variable. The gcv uses GCV instead of the residual sum of squares. Adding a variable can actually increase the GCV due to the model complexity penalty in the denominator.

> evimp(oz.mars,trim=FALSE)

nsubsets gcv rss

safb 11 100.0 100.0

inbh 10 48.9 50.8

hum 9 39.3 41.5

v500 8 32.8 35.1

day 7 29.5 31.7

dagg 5 22.2 24.2

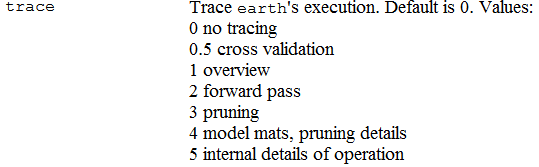
vis 4 13.4 16.6

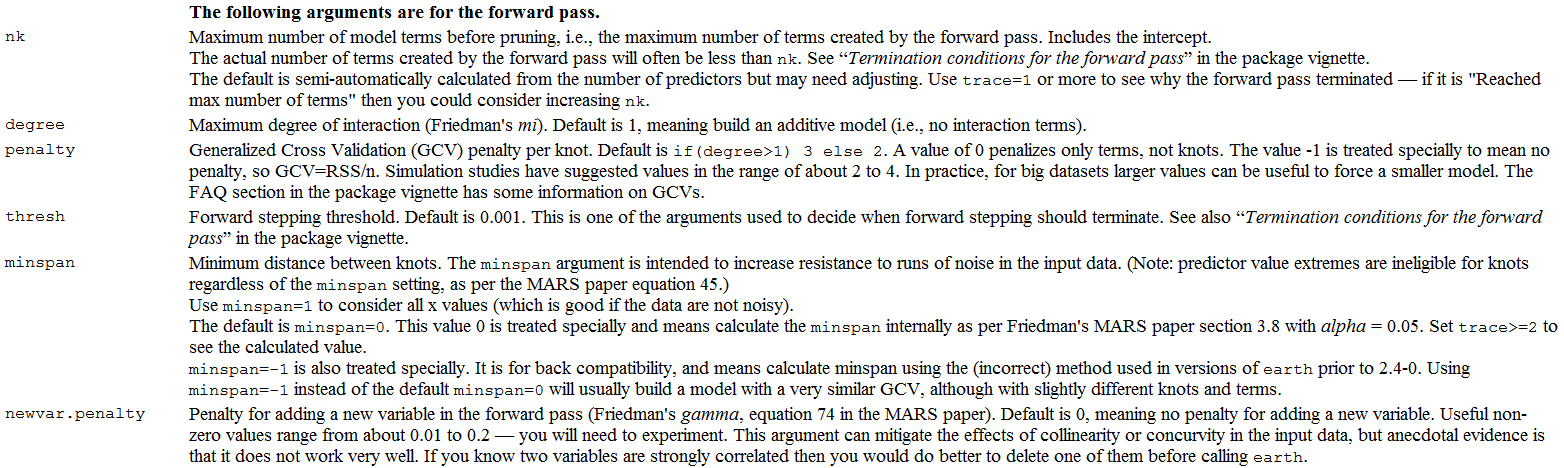
inbt 2 7.8 10.5

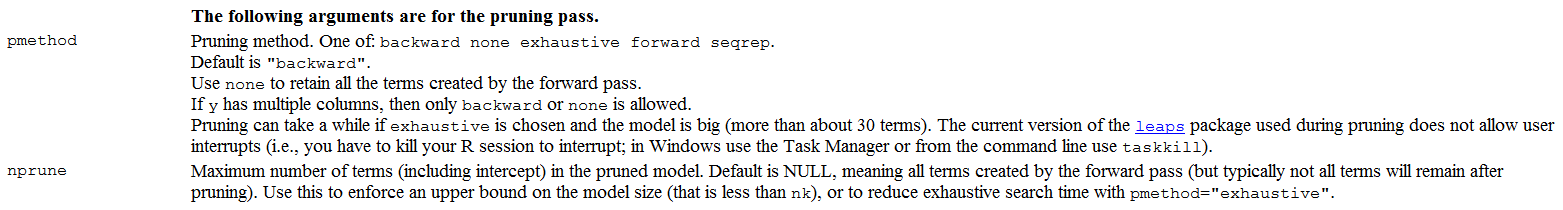
wind-unused 0 0.0 0.0

The trim = FALSE option displays variables not used in the model.

**Additional useful/interesting options for the earth() function**







**Example 5.2: Boston Housing Data**  
In this example we will use the same version of the Boston Housing data you used for Assignment 1. We will use a split-sample train the model with 400 observations in the training data, leaving 106 observations in the validation data set. We will then compare MARS models of varying complexities, with and without variable pre-processing.

> boston = read.table(file.choose(),header=T,sep=",")

> names(boston)

[1] "CMEDV" "CRIM" "ZN" "INDUS" "CHAS" "NOX" "RM" "AGE" "DIS" "RAD" "TAX"

[12] "PTRATIO" "B" "LSTAT"

> dim(boston)

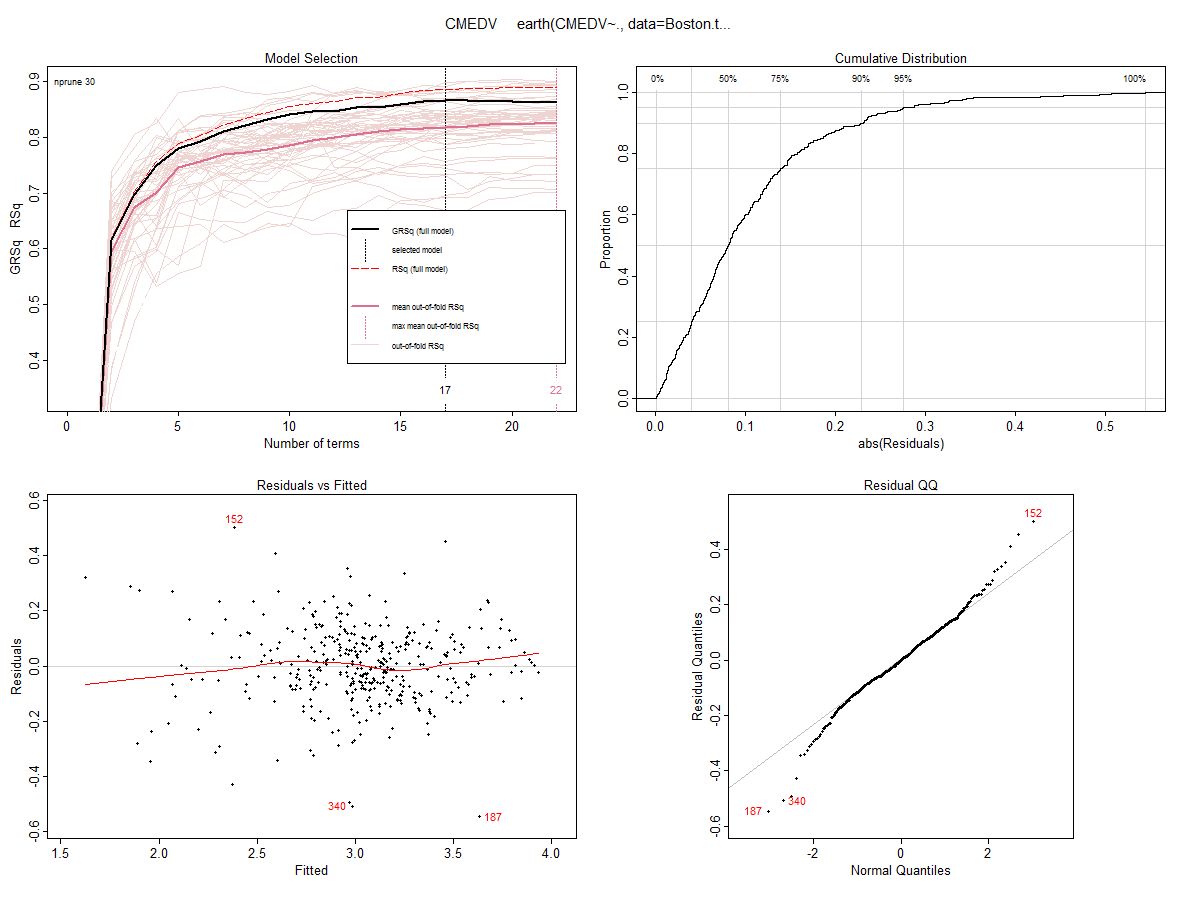
[1] 506 14

> train = sample(1:506,size=400,replace=F)  
  
> Boston.train = boston[train,]

> Boston.train$CMEDV=log(Boston.train$CMEDV)

> bos.mars = earth(CMEDV~.,data=Boston.train,degree=1,nfold=5,ncross=10,keepxy=T,nprune=30)

> bos.mars1 = earth(CMEDV~.,data=Boston.train,degree=1,nfold=5,ncross=10,keepxy=T,nprune=30)

> plot(bos.mars1)  


> summary(bos.mars1)

Call: earth(formula=CMEDV~., data=Boston.train, keepxy=T, degree=1, nprune=30, ncross=10, nfold=5)

coefficients

(Intercept) 21.8607447

h(CRIM-0.15086) -0.8953563

h(22.0511-CRIM) -0.8840920

h(CRIM-22.0511) 0.8904860

h(NOX-0.647) -12.1854042

h(NOX-0.693) 27.4802013

h(NOX-0.718) -15.6113612

h(5.854-RM) 0.1143655

h(RM-5.854) 0.1858658

h(1.5741-DIS) 0.8055974

h(TAX-296) 0.0025486

h(666-TAX) 0.0020154

h(TAX-666) -0.0133549

h(PTRATIO-14.7) -0.0330552

h(395.62-B) -0.0003657

h(6.21-LSTAT) 0.0626757

h(LSTAT-6.21) -0.0244515

Selected 17 of 22 terms, and 8 of 13 predictors

Termination condition: Reached nk 27

Importance: LSTAT, RM, CRIM, DIS, NOX, PTRATIO, TAX, B, ZN-unused, INDUS-unused, CHAS-unused, AGE-unused, ...

Number of terms at each degree of interaction: 1 16 (additive model)

GCV 0.02257949 RSS 7.603024 GRSq 0.8655558 RSq 0.886256 CVRSq 0.8198516

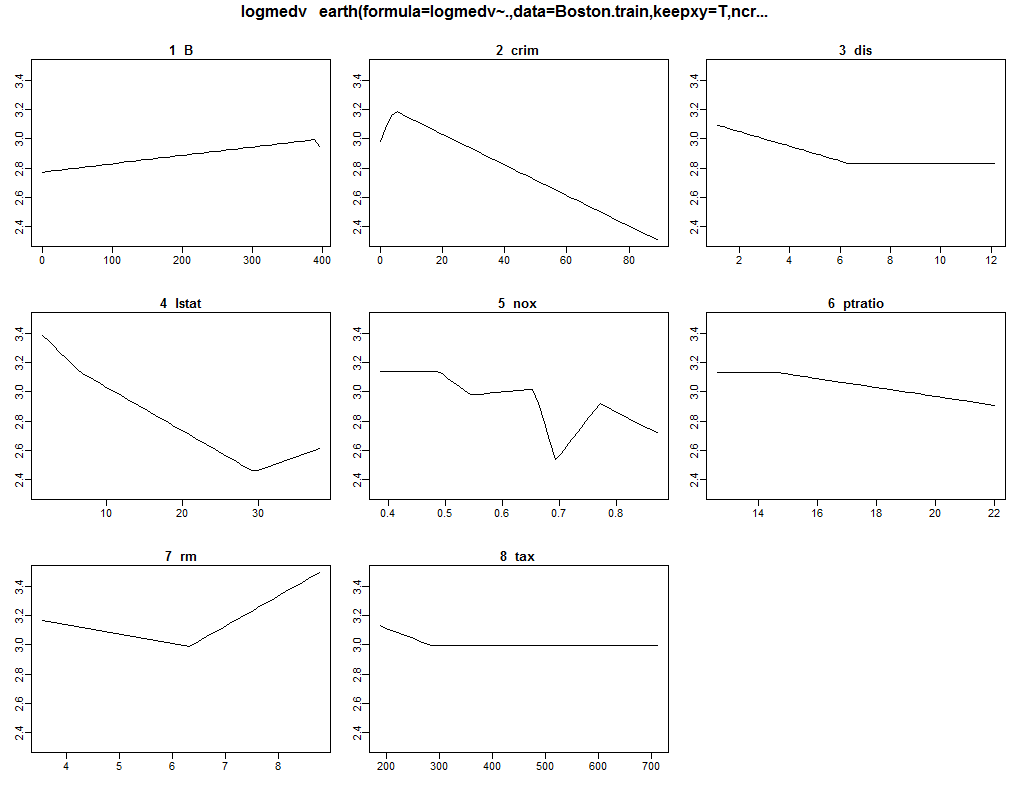
Note: the cross-validation sd's below are standard deviations across folds

Cross validation: nterms 18.16 sd 1.28 nvars 8.60 sd 0.90

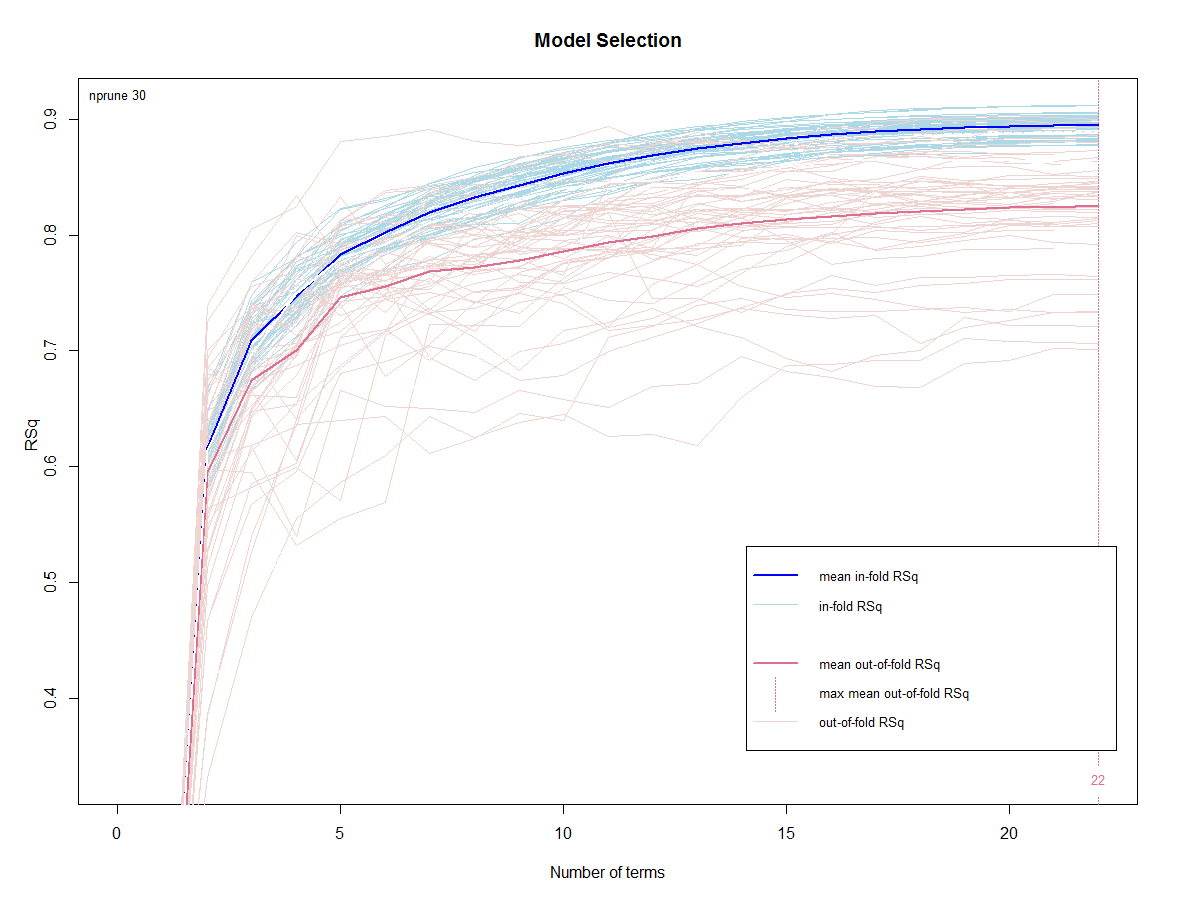
CVRSq sd MaxErr sd

0.82 0.051 1.03 0.62

> plotmo(bos.mars)



> plot(bos.mars,which=1,col.mean.infold.rsq="blue",col.infold.rsq="lightblue",col.grsq=0,col.rsq=0)



> Boston.test = boston[-train,]

> ypred.log = predict(bos.mars1,newdata=Boston.test)

> ypred = exp(ypred.log)

> RMSEP.test = sqrt(mean((Boston.test$CMEDV-ypred)^2))

> RMSEP.test

[1] 4.028772

> MAE.test = mean(abs(Boston.test$CMEDV-ypred))

> MAE.test

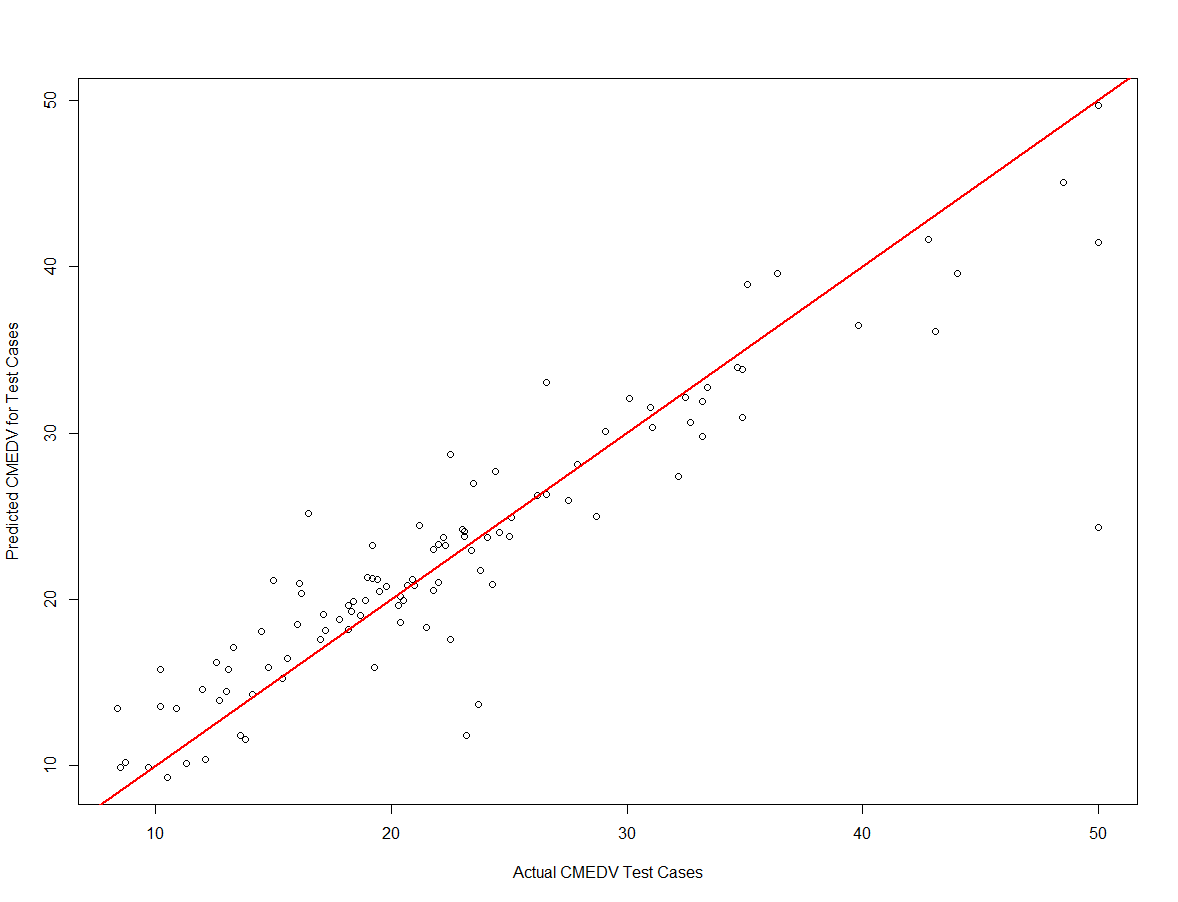
[1] 2.525717

> MAPE.test = mean(MAE.test/Boston.test$CMEDV)\*100

> MAPE.test

[1] 13.04505

> plot(Boston.test$CMEDV,ypred,xlab="Actual CMEDV Test Cases",ylab="Predicted CMEDV for Test Cases")

> abline(0,1,lwd=2,col="red")  


We now consider adding interaction terms to the MARS model.

> bos.mars2 = earth(CMEDV~.,data=Boston.train,degree=2,nfold=5,ncross=20,keepxy=T,nk=30)

> summary(bos.mars2)

Call: earth(formula=CMEDV~., data=Boston.train, keepxy=T, degree=2, ncross=20, nfold=5, nk=30)

coefficients

(Intercept) 2.63043953

h(22.0511-CRIM) 0.01239207

h(CRIM-22.0511) -0.00447799

h(5.854-RM) 0.18912219

h(RM-5.854) 0.23948843

h(98.1-AGE) 0.00165270

h(AGE-98.1) 0.03839334

h(243-TAX) 0.00470192

h(TAX-243) 0.00047014

h(20.2-PTRATIO) 0.05284040

h(6.21-LSTAT) 0.04899920

h(LSTAT-6.21) -0.02525114

h(22.0511-CRIM) \* h(1.5894-DIS) 0.11061290

h(22.0511-CRIM) \* h(393.77-B) -0.00002854

h(NOX-0.647) \* h(RM-5.854) -2.15234579

h(NOX-0.647) \* h(LSTAT-6.21) -0.51285808

h(NOX-0.693) \* h(LSTAT-6.21) 2.18838223

h(0.7-NOX) \* h(LSTAT-6.21) 0.07157907

h(NOX-0.7) \* h(LSTAT-6.21) -1.39198380

h(5.854-RM) \* h(TAX-666) -0.02459574

h(5.854-RM) \* h(666-TAX) -0.00083577

h(1.7984-DIS) \* h(20.2-PTRATIO) -0.53794918

h(DIS-1.7984) \* h(20.2-PTRATIO) -0.00903999

Selected 23 of 27 terms, and 9 of 13 predictors

Termination condition: Reached nk 30

Importance: LSTAT, RM, NOX, CRIM, DIS, PTRATIO, TAX, B, AGE, ZN-unused, INDUS-unused, CHAS-unused, RAD-unused

Number of terms at each degree of interaction: 1 11 11

GCV 0.01917629 RSS 5.673114 GRSq 0.8858193 RSq 0.9151282 CVRSq 0.690986

Note: the cross-validation sd's below are standard deviations across folds

Cross validation: nterms 20.57 sd 1.58 nvars 8.63 sd 0.77

CVRSq sd MaxErr sd

0.691 0.566 -6.91 1.32

> ypred.log = predict(bos.mars2,newdata=Boston.test)

> ypred = exp(ypred.log)

> PredAcc(Boston.test$CMEDV,ypred)

RMSEP

===============

3.464964

MAE

===============

2.515402

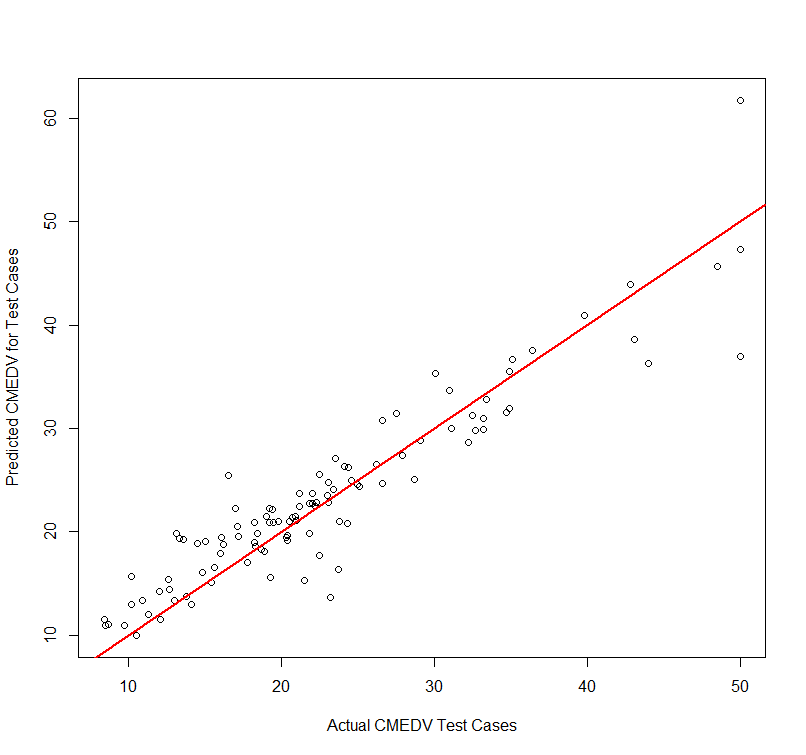
MAPE

===============

12.50276

> plot(Boston.test$CMEDV,ypred,xlab="Actual CMEDV Test Cases",ylab="Predicted CMEDV for Test Cases")

> abline(0,1,lwd=2,col="red")



650 3.61400 3.67700 88.98000

> boston.trans = boston

> boston.trans$CMEDV=log(boston$CMEDV)

> boston.trans$CRIM =log(boston$CRIM)

> boston.trans$LSTAT =log(boston$LSTAT)

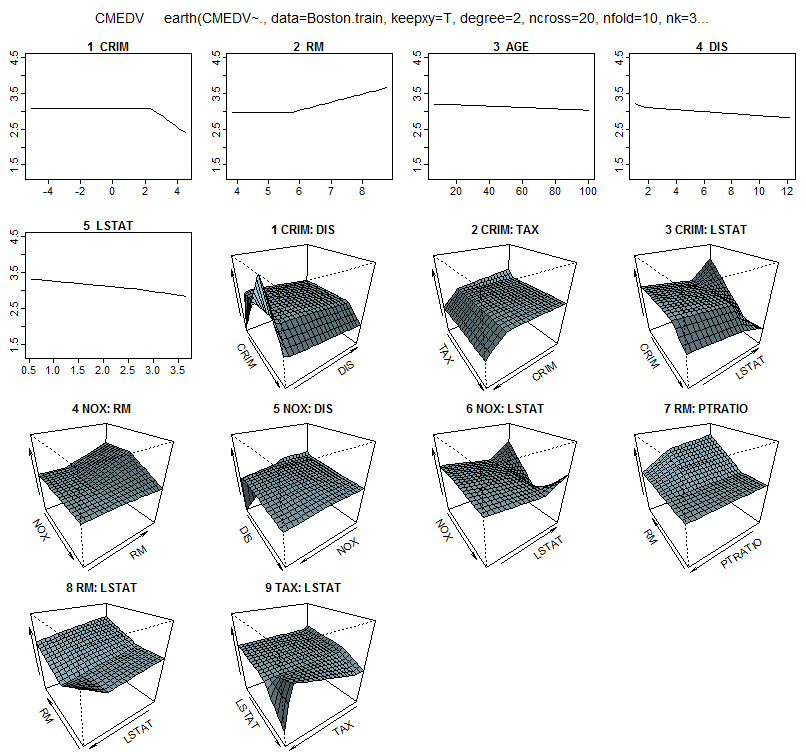
> boston.trans$ZN =log(boston$ZN+1)

> Boston.train = boston.trans[train,]

> Boston.test = boston.trans[-train,]

> bos.mars3 = earth(CMEDV~.,data=Boston.train,degree=2,ncross=20,nfold=10,nk=30,keepxy=T)

> plotmo(bos.mars3)



> summary(bos.mars3)

Call: earth(formula=CMEDV~., data=Boston.train, keepxy=T, degree=2, ncross=20, nfold=10, nk=30)

coefficients

(Intercept) 3.0795999

h(CRIM-2.41237) -0.3212208

h(RM-5.708) 0.2520660

h(AGE-21.4) -0.0018931

h(1.7523-DIS) 2.2969933

h(DIS-1.7523) -0.0279192

h(2.64617-LSTAT) 0.1323246

h(LSTAT-2.64617) -1.5034106

h(1.39585-CRIM) \* h(1.7523-DIS) -0.7872145

h(CRIM-1.39585) \* h(1.7523-DIS) -0.6622170

h(2.41237-CRIM) \* h(233-TAX) 0.0009265

h(2.29421-CRIM) \* h(LSTAT-2.64617) 0.2832995

h(CRIM-2.29421) \* h(LSTAT-2.64617) 0.3862097

h(NOX-0.624) \* h(RM-5.708) -1.1016751

h(NOX-0.597) \* h(1.7523-DIS) -10.9180523

h(0.693-NOX) \* h(LSTAT-2.64617) 5.7390533

h(NOX-0.693) \* h(LSTAT-2.64617) 8.8782185

h(RM-5.708) \* h(PTRATIO-18.6) -0.0664313

h(RM-5.708) \* h(18.6-PTRATIO) 0.0151582

h(5.708-RM) \* h(LSTAT-2.84258) 0.4735306

h(666-TAX) \* h(LSTAT-2.64617) -0.0018466

h(TAX-666) \* h(LSTAT-2.64617) -0.0354771

Selected 22 of 28 terms, and 8 of 13 predictors

Termination condition: Reached nk 30

Importance: LSTAT, RM, PTRATIO, NOX, DIS, CRIM, TAX, AGE, ZN-unused, INDUS-unused, CHAS-unused, RAD-unused, ...

Number of terms at each degree of interaction: 1 7 14

GCV 0.01790399 RSS 5.373983 GRSq 0.893395 RSq 0.9196033 CVRSq 0.757312

Note: the cross-validation sd's below are standard deviations across folds

Cross validation: nterms 20.95 sd 1.57 nvars 8.51 sd 0.77

CVRSq sd MaxErr sd

0.757 0.819 -7.01 0.717

> ypred.log = predict(bos.mars3,newdata=Boston.test)

> ypred = exp(ypred.log)

> PredAcc(exp(Boston.test$CMEDV),ypred)

RMSEP

===============

3.3846

MAE

===============

2.235294

MAPE

===============

11.15446

RMSEP MAE MAPE

1 3.3846 2.235294 11.15446

Compared to other methods we will examine later in the course we will see that MARS does not measure up. MARS is one of the main algorithms used in Salford Systems (<https://www.salford-systems.com/>) commercial predictive analytics software. Although MARS may not perform as well as some of the methods we will examine later in the course, it does produce a much interpretable model.   
  
In summary the MARS models are really just an MLR OLS model with automatically chosen terms and interactions defined by using a specific set of candidate basis terms as the “final” MARS model below demonstrates.

> summary(bos.mars3)

Call: earth(formula=CMEDV~., data=Boston.train, keepxy=T, degree=2, ncross=20, nfold=10, nk=30)

coefficients

(Intercept) 3.0795999

h(CRIM-2.41237) -0.3212208

h(RM-5.708) 0.2520660

h(AGE-21.4) -0.0018931

h(1.7523-DIS) 2.2969933

h(DIS-1.7523) -0.0279192

h(2.64617-LSTAT) 0.1323246

h(LSTAT-2.64617) -1.5034106

h(1.39585-CRIM) \* h(1.7523-DIS) -0.7872145

h(CRIM-1.39585) \* h(1.7523-DIS) -0.6622170

h(2.41237-CRIM) \* h(233-TAX) 0.0009265

h(2.29421-CRIM) \* h(LSTAT-2.64617) 0.2832995

h(CRIM-2.29421) \* h(LSTAT-2.64617) 0.3862097

h(NOX-0.624) \* h(RM-5.708) -1.1016751

h(NOX-0.597) \* h(1.7523-DIS) -10.9180523

h(0.693-NOX) \* h(LSTAT-2.64617) 5.7390533

h(NOX-0.693) \* h(LSTAT-2.64617) 8.8782185

h(RM-5.708) \* h(PTRATIO-18.6) -0.0664313

h(RM-5.708) \* h(18.6-PTRATIO) 0.0151582

h(5.708-RM) \* h(LSTAT-2.84258) 0.4735306

h(666-TAX) \* h(LSTAT-2.64617) -0.0018466

h(TAX-666) \* h(LSTAT-2.64617) -0.0354771

Fitted MARS model for pre-processed (i.e. transformations used) Boston Housing data expressed as a MLR model (i.e. ).