Regresión Spline adaptativa multivariante (MARS)

Víctor Morales Oñate
Sitio personal
ResearchGate
GitHub
LinkedIn

07 de mayo de 2019

Contents

Paquetes de esta sección

```
if(!require(ISLR)){install.packages("earth")}
if(!require(ISLR)){install.packages("caret")}
if(!require(ISLR)){install.packages("AmesHousing")}
```

- En las clases pasadas hemos revisado extensiones de la regresión linear (nls, regresión polinómica, entre otras).
- Existen otras variaciones como la regresión *Ridge*, *LASSO* y *Elastic NET* (algunas se verán en el módulo de Aprendizaje Automático).

Introducción

- En estadística, MARS es una forma de regresión lineal introducida por Jorome Friedan en 1991.
- MARS es una técnica de regresión no paramétrica y puede ser vista como una extensión de los modelos lineales que automáticamente no linealidades e interacciones entre variables.
- El término *MARS* está protegido por derechos de autor y pertenece a *Salford Systems*.
- Para evitar violentar esos derechos, las implementaciones abiertas de MARS se suelen llamar *Earth* (El paquete `earth` en R, por ejemplo).
- La Regresión Spline adaptativa multivariante (*Multivariate adaptive regression spline* - MARS)

¿Por qué usar modelos MARS?

- MARS es ideal para usuarios que prefieren obtener resultados similares a la regresión tradicional mientras capturan no linealidades e interacciones necesarias.
- MARS revela patrones importantes en los datos que otras técnicas suelen fallar en revelar.
- MARS construye su modelo uniendo pedazos de líneas rectas que mantienen su propia pendiente.

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲ *= good,* ◆ *=fair, and* ▼ *=poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

Figure 1: Fuente: ESLII

- Esto permite que se detecte cualquier patrón en los datos.
- Se puede utilizar para cuando se tiene variables de respuesta cuantitativa y cualitativa.
- MARS realiza (todo automático y con gran velocidad):
  - selección de variables.
  - transformación de variables.
  - detección de interacciones.
  - testeo

Áreas donde ha mostrado ser una técnica exitosa

- Predicción de demanda de electricidad de companías generadoras.
- Relacionar puntajes de satisfacción del cliente con las especificaciones técnicas del producto.
- Modelización en sistemas de información geográfica.
- MARS es una técnica de regresión muy versátil y es una herramienta necesaria en nuestra caja de herramientas en Analítica de Datos.

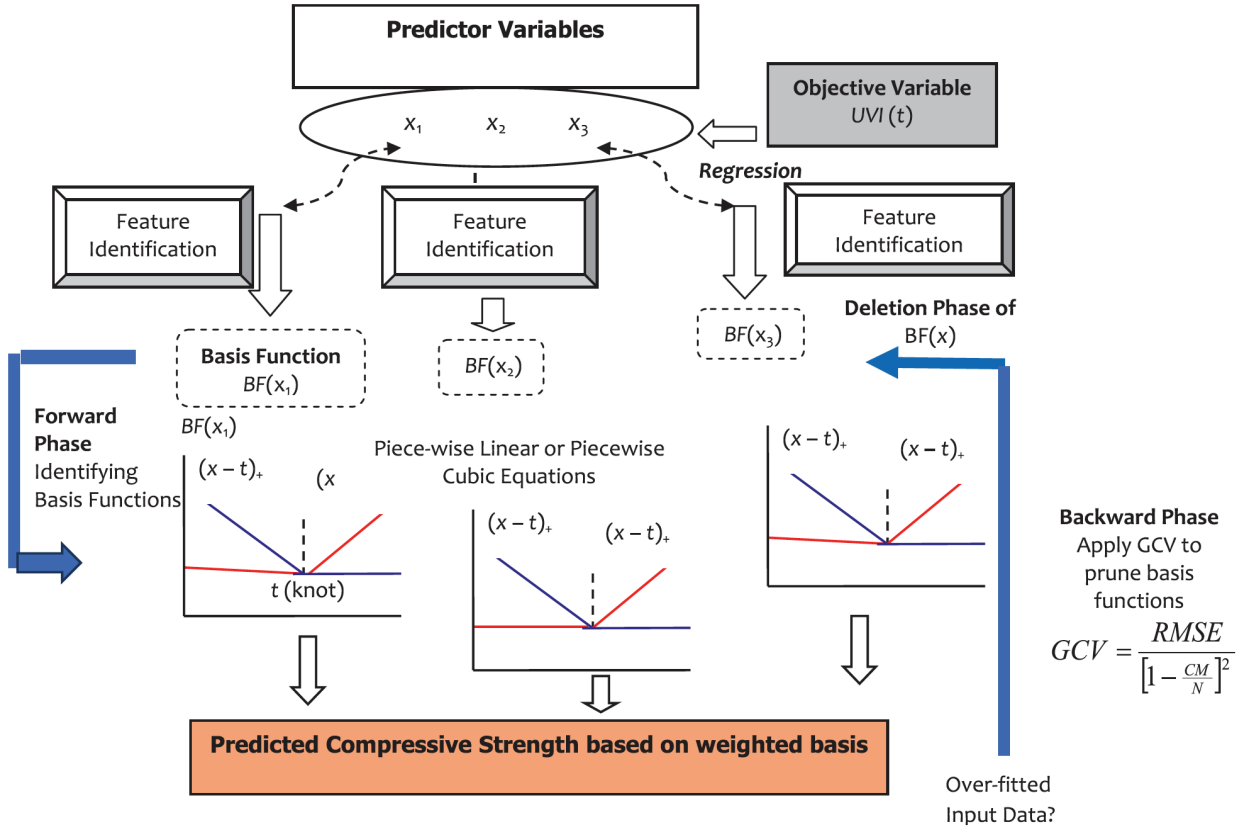


Fig. 2. Multivariate adaptive regression splines model architecture.

Figure 2: Fuente: Yaseena2018

La estrucura de MARS

Ejemplo 1

Cargamos los datos:

```
library(earth)
load("~/Documents/Consultorias&Cursos/DataLectures/banckfull.RData")
```

Construimos el modelo basado en los datos:

```
mars <- earth(y~age+job+marital+education+default+balance+housing+
loan+contact+day+month+duration+campaign+pdays+previous+poutcome,
data=bankfull,pmethod="backward",nprune=20, nfold=10)
```

Notemos los argumentos usando en la función:

- `pmethod`: Es el método para podar las variables regresoras. Las opciones son `backward`, `forward`, `cv` (se necesita especificar `nfold`), y `exhaustive`.
- `nprune`: Numero máximo de funciones base que se usan.

En resumen, para plantear el modelo, necesitamos 3 elementos:

1. Definir el modelo (como en cualquier regresión)
2. Definir el método de testeo (`pmethod`)

3. Número de funciones base (`nprune`) y de interacciones (`degree`)

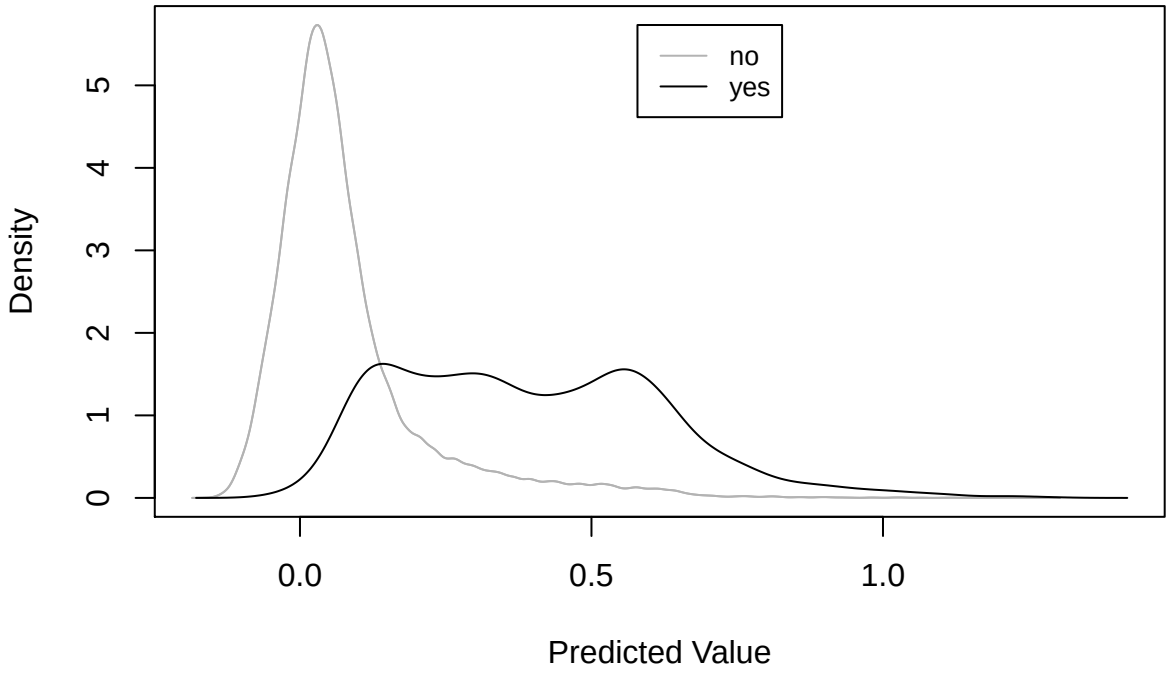
Veamos el resumen:

```
summary(mars,digit=3)
```

```
## Call:
##             earth(formula=y~age+job+marital+education+default+balance+housin...),
##             data=bankfull, pmethod="backward", nprune=20, nfold=10)
##
##                   coefficients
## (Intercept)            0.7775
## housingyes            -0.0408
## loanyes               -0.0294
## contactunknown        -0.0713
## monthdec               0.1876
## monthjun               0.0519
## monthmar               0.3301
## monthoct               0.1916
## monthsep               0.1789
## poutcomesuccess        0.3809
## h(age-27)              0.0072
## h(54-age)              0.0087
## h(duration-375)        0.0003
## h(1080-duration)      -0.0004
## h(duration-1080)      -0.0004
## h(2-campaign)          0.0268
## h(pdays-53)           -0.0020
## h(349-pdays)          -0.0016
## h(pdays-349)           0.0061
## h(pdays-425)          -0.0044
##
## Selected 20 of 22 terms, and 13 of 42 predictors
## Termination condition: RSq changed by less than 0.001 at 22 terms
## Importance: duration, poutcomesuccess, monthmar, housingyes, monthoct, ...
## Number of terms at each degree of interaction: 1 19 (additive model)
## GCV 0.0707  RSS 3192  GRSq 0.315  RSq 0.316  CVRSq 0.316
##
## Note: the cross-validation sd's below are standard deviations across folds
##
## Cross validation:   nterms 22.60 sd 1.35    nvars 14.20 sd 1.99
##
##      CVRSq    sd     ClassRate    sd     MaxErr   sd
##      0.316 0.019         0.901 0.003       -1.3 1.13
```

El gráfico de resultado:

```
plotd(mars)
```



El GCV (generalized cross validation) es

$$GCV = \frac{RSS}{N \times (1 - Num.Par.Efectivos/N)^2}$$

donde RSS es la suma de cuadrados de los residuos medidos en los datos de entrenamiento y N es el número de observaciones.

$$Num.Par.Efectivos = NumeroTerminosMARS + Penalidad \times (NumeroTerminosMARS - 1)/2$$

La penalidad es alrededor de 2 o 3, pero se puede elegir la penalidad.

Output

El objeto de resultado es un `earth.object` que contiene mucha información (ver `help(earth.object)`.

```
str(mars)
```

```
## List of 39
##  $ rss                      : num 3192
##  $ rsq                      : num 0.316
##  $ gcv                      : num 0.0707
##  $ grsq                     : num 0.315
##  $ bx                       : num [1:45211, 1:20] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:20] "(Intercept)" "h(duration-1080)" "h(1080-duration)" "poutcomesuccess" ...
##  $ dirs                     : num [1:22, 1:42] 0 0 0 0 0 0 0 0 0 1 ...
```

```
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:22] "(Intercept)" "h(duration-1080)" "h(1080-duration)" "poutcomesuccess" ...
##   .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##  $ cuts                    : num [1:22, 1:42] 0 0 0 0 0 0 0 0 0 0 54 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:22] "(Intercept)" "h(duration-1080)" "h(1080-duration)" "poutcomesuccess" ...
##   .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##  $ selected.terms          : num [1:20] 1 2 3 4 5 6 7 8 9 11 ...
##  $ prune.terms             : num [1:20, 1:20] 1 1 1 1 1 1 1 1 1 1 ...
##  $ fitted.values           : num [1:45211, 1] 0.0261 -0.0314 -0.074 -0.0597 0.0452 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "yes"
##  $ residuals               : num [1:45211, 1] -0.0261 0.0314 0.074 0.0597 -0.0452 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "yes"
##  $ coefficients            : num [1:20, 1] 0.777457 -0.000382 -0.000402 0.380944 0.330111 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:20] "(Intercept)" "h(duration-1080)" "h(1080-duration)" "poutcomesuccess" ...
##   .. ..$ : chr "yes"
##  $ rss.per.response        : num 3192
##  $ rsq.per.response        : num 0.316
##  $ gcv.per.response        : num 0.0707
##  $ grsq.per.response       : num 0.315
##  $ rss.per.subset          : num [1:20] 4670 3880 3497 3433 3378 ...
##  $ gcv.per.subset          : num [1:20] 0.1033 0.0858 0.0774 0.076 0.0747 ...
##  $ leverages               : num [1:45211] 0.000243 0.000165 0.000299 0.000194 0.00025 ...
##  $ pmethod                 : chr "backward"
##  $ nprune                  : num 20
##  $ penalty                 : num 2
##  $ nk                      : num 85
##  $ thresh                  : num 0.001
##  $ termcond                : int 4
##  $ weights                 : NULL
##  $ call                    : language earth(formula = y ~ age + job + marital + education + default
##  $ namesx.org              : chr [1:16] "age" "job" "marital" "education" ...
##  $ namesx                  : chr [1:16] "age" "job" "marital" "education" ...
##  $ terms                   :Classes 'terms', 'formula'  language y ~ age + job + marital + educatio
##   .. ..- attr(*, "variables")= language list(y, age, job, marital, education, default, balance, hous
##   .. ..- attr(*, "factors")= int [1:17, 1:16] 0 1 0 0 0 0 0 0 0 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:17] "y" "age" "job" "marital" ...
##   .. .. .. ..$ : chr [1:16] "age" "job" "marital" "education" ...
##   .. ..- attr(*, "term.labels")= chr [1:16] "age" "job" "marital" "education" ...
##   .. ..- attr(*, "order")= int [1:16] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(y, age, job, marital, education, default, balance, housi
##   .. ..- attr(*, "dataClasses")= Named chr [1:17] "factor" "numeric" "factor" "factor" ...
##   .. .. ..- attr(*, "names")= chr [1:17] "y" "age" "job" "marital" ...
##  $ xlevels                 :List of 9
##   ..$ job      : chr [1:12] "admin." "blue-collar" "entrepreneur" "housemaid" ...
```

```
##   ..$ marital  : chr [1:3] "divorced" "married" "single"
##   ..$ education: chr [1:4] "primary" "secondary" "tertiary" "unknown"
##   ..$ default  : chr [1:2] "no" "yes"
##   ..$ housing  : chr [1:2] "no" "yes"
##   ..$ loan     : chr [1:2] "no" "yes"
##   ..$ contact  : chr [1:3] "cellular" "telephone" "unknown"
##   ..$ month    : chr [1:12] "apr" "aug" "dec" "feb" ...
##   ..$ poutcome : chr [1:4] "failure" "other" "success" "unknown"
##  $ levels                   : chr [1:2] "no" "yes"
##  $ cv.list                  :List of 10
##   ..$ fold1 :List of 29
##   .. ..$ rss              : num 2880
##   .. ..$ rsq              : num 0.315
##   .. ..$ gcv              : num 0.0709
##   .. ..$ grsq             : num 0.314
##   .. ..$ dirs             : num [1:22, 1:42] 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:22] "(Intercept)" "h(duration-1133)" "h(1133-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:22, 1:42] 0 0 0 0 0 0 0 0 0 51 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:22] "(Intercept)" "h(duration-1133)" "h(1133-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:21] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40721, 1] 0.0326 -0.0346 -0.0788 -0.0627 0.0405 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:21, 1] 0.568207 -0.000404 -0.000406 0.384036 -0.040421 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:21] "(Intercept)" "h(duration-1133)" "h(1133-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2880
##   .. ..$ rsq.per.response : num 0.315
##   .. ..$ gcv.per.response : num 0.0709
##   .. ..$ grsq.per.response: num 0.314
##   .. ..$ rss.per.subset   : num [1:22] 4204 3493 3148 3091 3042 ...
##   .. ..$ gcv.per.subset   : num [1:22] 0.1032 0.0858 0.0773 0.0759 0.0747 ...
##   .. ..$ leverages        : num [1:40721] 0.000282 0.000189 0.000341 0.000229 0.000287 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 1
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold2 :List of 29
```

```
##   .. ..$ rss              : num 2869
##   .. ..$ rsq              : num 0.317
##   .. ..$ gcv              : num 0.0707
##   .. ..$ grsq             : num 0.316
##   .. ..$ dirs             : num [1:25, 1:42] 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:25] "(Intercept)" "h(duration-1080)" "h(1080-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:25, 1:42] 0 0 0 0 0 0 0 0 0 55 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:25] "(Intercept)" "h(duration-1080)" "h(1080-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:23] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40675, 1] 0.0377 -0.0353 -0.0519 -0.0468 0.0336 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:23, 1] 1.007423 -0.000381 -0.000394 0.381602 0.31842 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:23] "(Intercept)" "h(duration-1080)" "h(1080-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2869
##   .. ..$ rsq.per.response : num 0.317
##   .. ..$ gcv.per.response : num 0.0707
##   .. ..$ grsq.per.response: num 0.316
##   .. ..$ rss.per.subset   : num [1:25] 4203 3491 3149 3091 3044 ...
##   .. ..$ gcv.per.subset   : num [1:25] 0.1033 0.0858 0.0774 0.076 0.0749 ...
##   .. ..$ leverages        : num [1:40675] 0.000311 0.000188 0.000219 0.000228 0.000289 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 2
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold3 :List of 29
##   .. ..$ rss              : num 2847
##   .. ..$ rsq              : num 0.323
##   .. ..$ gcv              : num 0.0701
##   .. ..$ grsq             : num 0.321
##   .. ..$ dirs             : num [1:27, 1:42] 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:27] "(Intercept)" "h(duration-1093)" "h(1093-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:27, 1:42] 0 0 0 0 0 0 0 0 0 53 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:27] "(Intercept)" "h(duration-1093)" "h(1093-duration)" "poutcomesuccess" ..
```

```
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:25] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40690, 1] 0.0221 -0.0392 -0.0563 -0.0661 0.0415 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:25, 1] 1.007298 -0.000386 -0.000397 0.376748 0.294318 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:25] "(Intercept)" "h(duration-1093)" "h(1093-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2847
##   .. ..$ rsq.per.response : num 0.323
##   .. ..$ gcv.per.response : num 0.0701
##   .. ..$ grsq.per.response: num 0.321
##   .. ..$ rss.per.subset   : num [1:27] 4203 3499 3148 3089 3038 ...
##   .. ..$ gcv.per.subset   : num [1:27] 0.1033 0.086 0.0774 0.076 0.0747 ...
##   .. ..$ leverages        : num [1:40690] 0.000282 0.000194 0.000226 0.00023 0.000302 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 3
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold4 :List of 29
##   .. ..$ rss              : num 2851
##   .. ..$ rsq              : num 0.322
##   .. ..$ gcv              : num 0.0703
##   .. ..$ grsq             : num 0.32
##   .. ..$ dirs             : num [1:22, 1:42] 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:22] "(Intercept)" "h(duration-1081)" "h(1081-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:22, 1:42] 0 0 0 0 0 0 0 0 0 55 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:22] "(Intercept)" "h(duration-1081)" "h(1081-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:21] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40660, 1] -0.0298 -0.0772 -0.0565 0.0419 -0.0249 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:21, 1] 0.353636 -0.000339 -0.000398 0.3808 0.333208 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:21] "(Intercept)" "h(duration-1081)" "h(1081-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2851
```

```
##   .. ..$ rsq.per.response : num 0.322
##   .. ..$ gcv.per.response : num 0.0703
##   .. ..$ grsq.per.response: num 0.32
##   .. ..$ rss.per.subset   : num [1:22] 4203 3480 3132 3073 3023 ...
##   .. ..$ gcv.per.subset   : num [1:22] 0.1034 0.0856 0.077 0.0756 0.0744 ...
##   .. ..$ leverages        : num [1:40660] 0.000183 0.000342 0.000216 0.000287 0.000185 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 4
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold5 :List of 29
##   .. ..$ rss              : num 2857
##   .. ..$ rsq              : num 0.32
##   .. ..$ gcv              : num 0.0704
##   .. ..$ grsq             : num 0.319
##   .. ..$ dirs             : num [1:25, 1:42] 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:25] "(Intercept)" "h(duration-1073)" "h(1073-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:25, 1:42] 0 0 0 0 0 0 0 0 0 55 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:25] "(Intercept)" "h(duration-1073)" "h(1073-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:24] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40680, 1] 0.0226 -0.0305 -0.0489 -0.0568 0.047 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:24, 1] 1.069982 -0.000394 -0.000394 0.380363 -0.047925 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:24] "(Intercept)" "h(duration-1073)" "h(1073-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2857
##   .. ..$ rsq.per.response : num 0.32
##   .. ..$ gcv.per.response : num 0.0704
##   .. ..$ grsq.per.response: num 0.319
##   .. ..$ rss.per.subset   : num [1:25] 4203 3498 3151 3096 3048 ...
##   .. ..$ gcv.per.subset   : num [1:25] 0.1033 0.086 0.0775 0.0761 0.075 ...
##   .. ..$ leverages        : num [1:40680] 0.000264 0.000178 0.00021 0.00021 0.000287 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
```

```
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 5
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold6 :List of 29
##   .. ..$ rss              : num 2875
##   .. ..$ rsq              : num 0.316
##   .. ..$ gcv              : num 0.0708
##   .. ..$ grsq             : num 0.315
##   .. ..$ dirs             : num [1:22, 1:42] 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:22] "(Intercept)" "h(duration-1076)" "h(1076-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:22, 1:42] 0 0 0 0 0 0 0 0 0 0 54 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:22] "(Intercept)" "h(duration-1076)" "h(1076-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:21] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40679, 1] 0.0235 -0.0369 -0.0583 -0.0634 0.0327 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:21, 1] 0.688727 -0.000382 -0.000407 0.385472 -0.041368 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:21] "(Intercept)" "h(duration-1076)" "h(1076-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2875
##   .. ..$ rsq.per.response : num 0.316
##   .. ..$ gcv.per.response : num 0.0708
##   .. ..$ grsq.per.response: num 0.315
##   .. ..$ rss.per.subset   : num [1:22] 4203 3490 3153 3095 3044 ...
##   .. ..$ gcv.per.subset   : num [1:22] 0.1033 0.0858 0.0775 0.0761 0.0749 ...
##   .. ..$ leverages        : num [1:40679] 0.000269 0.000177 0.000223 0.000211 0.000295 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 6
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold7 :List of 29
##   .. ..$ rss              : num 2865
```

```
##   .. ..$ rsq              : num 0.318
##   .. ..$ gcv              : num 0.0706
##   .. ..$ grsq             : num 0.317
##   .. ..$ dirs             : num [1:24, 1:42] 0 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:24] "(Intercept)" "h(duration-1084)" "h(1084-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:24, 1:42] 0 0 0 0 0 0 0 0 0 0 55 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:24] "(Intercept)" "h(duration-1084)" "h(1084-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:23] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40698, 1] 0.0373 -0.0419 -0.0617 -0.0511 0.0272 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:23, 1] 0.458865 -0.000394 -0.000406 0.366318 0.33141 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:23] "(Intercept)" "h(duration-1084)" "h(1084-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2865
##   .. ..$ rsq.per.response : num 0.318
##   .. ..$ gcv.per.response : num 0.0706
##   .. ..$ grsq.per.response: num 0.317
##   .. ..$ rss.per.subset   : num [1:24] 4203 3495 3152 3094 3044 ...
##   .. ..$ gcv.per.subset   : num [1:24] 0.1033 0.0859 0.0775 0.076 0.0748 ...
##   .. ..$ leverages        : num [1:40698] 0.000318 0.000187 0.000221 0.000229 0.000293 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 7
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold8 :List of 29
##   .. ..$ rss              : num 2880
##   .. ..$ rsq              : num 0.315
##   .. ..$ gcv              : num 0.0709
##   .. ..$ grsq             : num 0.313
##   .. ..$ dirs             : num [1:23, 1:42] 0 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:23] "(Intercept)" "h(duration-1074)" "h(1074-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:23, 1:42] 0 0 0 0 0 0 0 0 0 0 53 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:23] "(Intercept)" "h(duration-1074)" "h(1074-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
```

```
##   .. ..$ selected.terms   : num [1:22] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40701, 1] 0.0427 -0.0427 -0.0578 0.0318 -0.0328 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:22, 1] 0.385863 -0.00038 -0.000395 0.380285 -0.041394 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:22] "(Intercept)" "h(duration-1074)" "h(1074-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2880
##   .. ..$ rsq.per.response : num 0.315
##   .. ..$ gcv.per.response : num 0.0709
##   .. ..$ grsq.per.response: num 0.313
##   .. ..$ rss.per.subset   : num [1:23] 4203 3493 3157 3099 3050 ...
##   .. ..$ gcv.per.subset   : num [1:23] 0.1033 0.0858 0.0776 0.0762 0.075 ...
##   .. ..$ leverages        : num [1:40701] 0.000331 0.00019 0.000223 0.000292 0.000182 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 8
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold9 :List of 29
##   .. ..$ rss              : num 2857
##   .. ..$ rsq              : num 0.32
##   .. ..$ gcv              : num 0.0704
##   .. ..$ grsq             : num 0.319
##   .. ..$ dirs             : num [1:24, 1:42] 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:24] "(Intercept)" "h(duration-1081)" "h(1081-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:24, 1:42] 0 0 0 0 0 0 0 0 0 55 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:24] "(Intercept)" "h(duration-1081)" "h(1081-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:23] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40695, 1] 0.0334 -0.0436 -0.0511 0.0267 -0.0355 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:23, 1] 0.464645 -0.000386 -0.000388 0.37834 0.332264 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:23] "(Intercept)" "h(duration-1081)" "h(1081-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2857
##   .. ..$ rsq.per.response : num 0.32
```

```
##   .. ..$ gcv.per.response : num 0.0704
##   .. ..$ grsq.per.response: num 0.319
##   .. ..$ rss.per.subset   : num [1:24] 4203 3491 3138 3080 3031 ...
##   .. ..$ gcv.per.subset   : num [1:24] 0.1033 0.0858 0.0771 0.0757 0.0745 ...
##   .. ..$ leverages        : num [1:40695] 0.000332 0.000187 0.000234 0.000294 0.000182 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 9
##   .. ..- attr(*, "class")= chr "earth"
##   ..$ fold10:List of 29
##   .. ..$ rss              : num 2858
##   .. ..$ rsq              : num 0.32
##   .. ..$ gcv              : num 0.0704
##   .. ..$ grsq             : num 0.319
##   .. ..$ dirs             : num [1:24, 1:42] 0 0 0 0 0 0 0 0 0 0 1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:24] "(Intercept)" "h(duration-1073)" "h(1073-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ cuts             : num [1:24, 1:42] 0 0 0 0 0 0 0 0 0 0 53 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:24] "(Intercept)" "h(duration-1073)" "h(1073-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ selected.terms   : num [1:23] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ fitted.values    : num [1:40700, 1] 0.04119 -0.06102 -0.05537 -0.03574 0.00699 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr "yes"
##   .. ..$ coefficients     : num [1:23, 1] 0.603174 -0.000391 -0.0004 0.370876 0.339198 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:23] "(Intercept)" "h(duration-1073)" "h(1073-duration)" "poutcomesuccess" ..
##   .. .. .. ..$ : chr "yes"
##   .. ..$ rss.per.response : num 2858
##   .. ..$ rsq.per.response : num 0.32
##   .. ..$ gcv.per.response : num 0.0704
##   .. ..$ grsq.per.response: num 0.319
##   .. ..$ rss.per.subset   : num [1:24] 4204 3494 3148 3086 3037 ...
##   .. ..$ gcv.per.subset   : num [1:24] 0.1033 0.0859 0.0774 0.0759 0.0747 ...
##   .. ..$ leverages        : num [1:40700] 0.000318 0.000222 0.000233 0.000182 0.000218 ...
##   .. ..$ pmethod          : chr "backward"
##   .. ..$ nprune           : NULL
##   .. ..$ penalty          : num 2
##   .. ..$ nk               : num 85
##   .. ..$ thresh           : num 0.001
##   .. ..$ termcond         : int 4
```

```
##   .. ..$ weights          : NULL
##   .. ..$ call             : language earth(x = infold.x, y = infold.y, weights = infold.weights, wp =
##   .. ..$ namesx.org       : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ namesx           : chr [1:42] "age" "jobblue-collar" "jobentrepreneur" "jobhousemaid" ...
##   .. ..$ levels           : num [1:2] 0 1
##   .. ..$ icross           : int 1
##   .. ..$ ifold            : int 10
##   .. ..- attr(*, "class")= chr "earth"
##  $ cv.nterms.selected.by.gcv: Named num [1:11] 21 23 25 21 24 21 23 22 23 23 ...
##   ..- attr(*, "names")= chr [1:11] "fold1" "fold2" "fold3" "fold4" ...
##  $ cv.nvars.selected.by.gcv : Named num [1:11] 13 16 18 13 17 13 13 13 13 13 ...
##   ..- attr(*, "names")= chr [1:11] "fold1" "fold2" "fold3" "fold4" ...
##  $ cv.groups               : int [1:45211, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "cross" "fold"
##  $ cv.rsq.tab              : num [1:11, 1:2] 0.331 0.333 0.311 0.276 0.324 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:11] "fold1" "fold2" "fold3" "fold4" ...
##   .. ..$ : chr [1:2] "yes" "mean"
##  $ cv.maxerr.tab           : num [1:11, 1:2] 1.06 -1.1 1.06 -1.11 -1.3 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:11] "fold1" "fold2" "fold3" "fold4" ...
##   .. ..$ : chr [1:2] "yes" "max"
##  $ cv.class.rate.tab       : num [1:11, 1:2] 0.901 0.904 0.899 0.897 0.903 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "yes" "mean"
##  - attr(*, "class")= chr "earth"
```

De todos este conjunto, vamos a destacar 3 elementos

1. Importancia de las variables
2. Funciones base (modelo resultado)
3. Curvas y superficie (contribución)

Importancia de las variables

```
library(caret)
varImp( mars  )
```

```
##                   Overall
## duration        100.000000
## poutcomesuccess  68.109084
## monthmar         45.171762
## housingyes       40.087272
## monthoct         35.114270
## contactunknown   31.401977
## monthsep         27.823303
## age              24.185852
## monthjun         21.090675
## pdays            16.010587
## monthdec         14.461722
## campaign         12.631608
## loanyes           5.779968
## job               0.000000
```

```
## marital           0.000000
## education         0.000000
## default           0.000000
## balance           0.000000
## housing           0.000000
## loan              0.000000
## contact           0.000000
## day               0.000000
## month             0.000000
## previous          0.000000
## poutcome          0.000000
```

Funciones Base

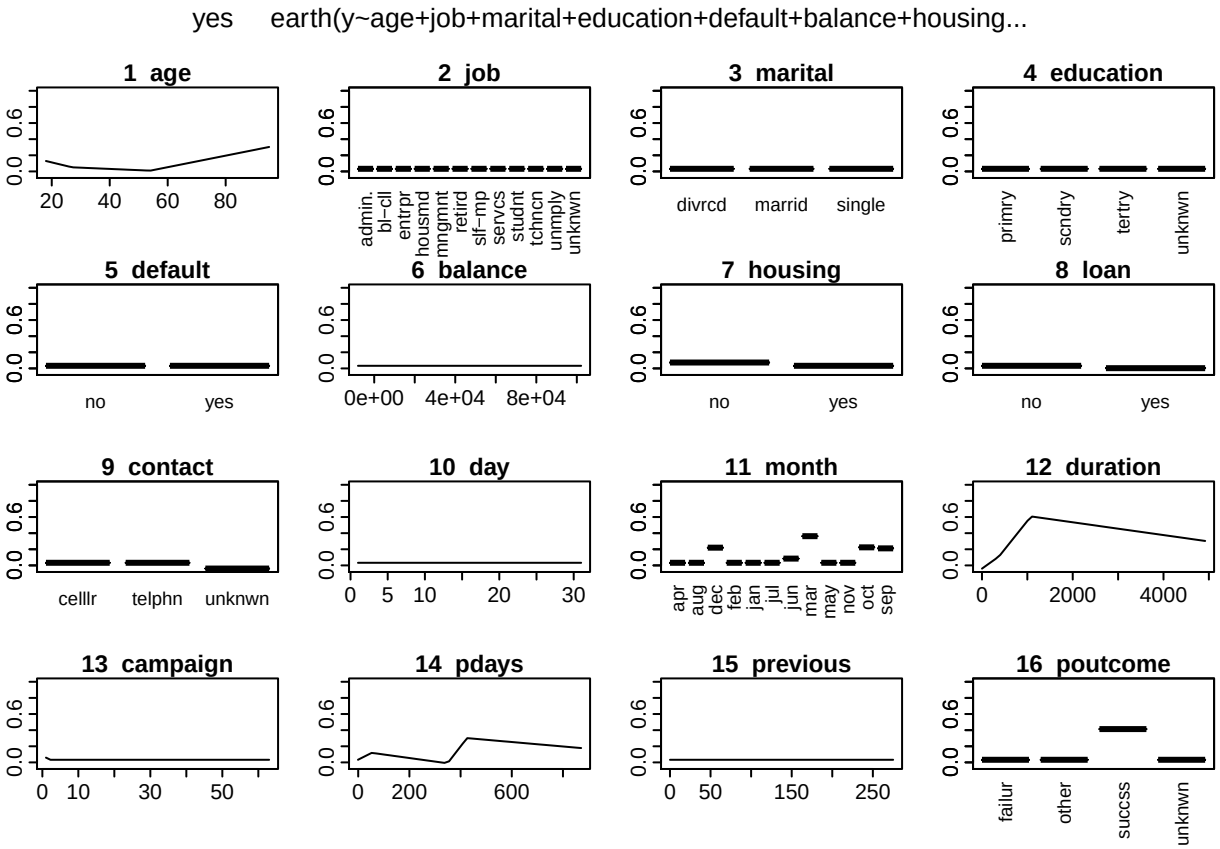
```
mars$coefficients
```

```
##                            yes
## (Intercept)        0.7774569240
## h(duration-1080) -0.0003818948
## h(1080-duration) -0.0004020631
## poutcomesuccess    0.3809444003
## monthmar           0.3301108826
## housingyes        -0.0407997273
## monthoct           0.1916481210
## contactunknown    -0.0712999709
## monthsep           0.1788583816
## h(54-age)          0.0087017318
## h(duration-375)    0.0003026388
## monthjun           0.0518693660
## h(2-campaign)      0.0268377535
## monthdec           0.1876019796
## h(pdays-349)       0.0061454449
## h(349-pdays)      -0.0015968138
## h(age-27)          0.0071639964
## h(pdays-53)       -0.0020353430
## h(pdays-425)      -0.0043865936
## loanyes           -0.0293712807
```

Curvas y superficie

```
plotmo( mars,  all1 = T   )
```

```
##  plotmo grid:    age         job marital education default balance housing
##                   39 blue-collar married secondary      no     448     yes
##  loan  contact day month duration campaign pdays previous poutcome
##    no cellular  16   may      180        2    -1        0  unknown
```



Tu turno

Sobre los datos `ames_train` ajusta un modelo MARS que tenga como variable dependiente al precio de venta `Sale_Price`.

```
library(rsample)
# Create training (70%) and test (30%) sets for the AmesHousing::make_ames() data.
# Use set.seed for reproducibility

set.seed(123)
ames_split <- initial_split(AmesHousing::make_ames(), prop = .7, strata = "Sale_Price")
ames_train <- training(ames_split)
ames_test  <- testing(ames_split)
```