Multiple regression and additive models

Patrick Breheny

November 29

Introduction

- Thus far, we have discussed nonparametric regression involving a single covariate
- In practice, we often have a p-dimensional vector of covariates for each observation
- The nonparametric multiple regression problem is therefore to estimate

$$\mathbb{E}(y|\mathbf{x}) = f(\mathbf{x})$$

  where $f : \mathbb{R}^p \mapsto \mathbb{R}$
- Both local regression methods and splines can be extended to deal with this problem

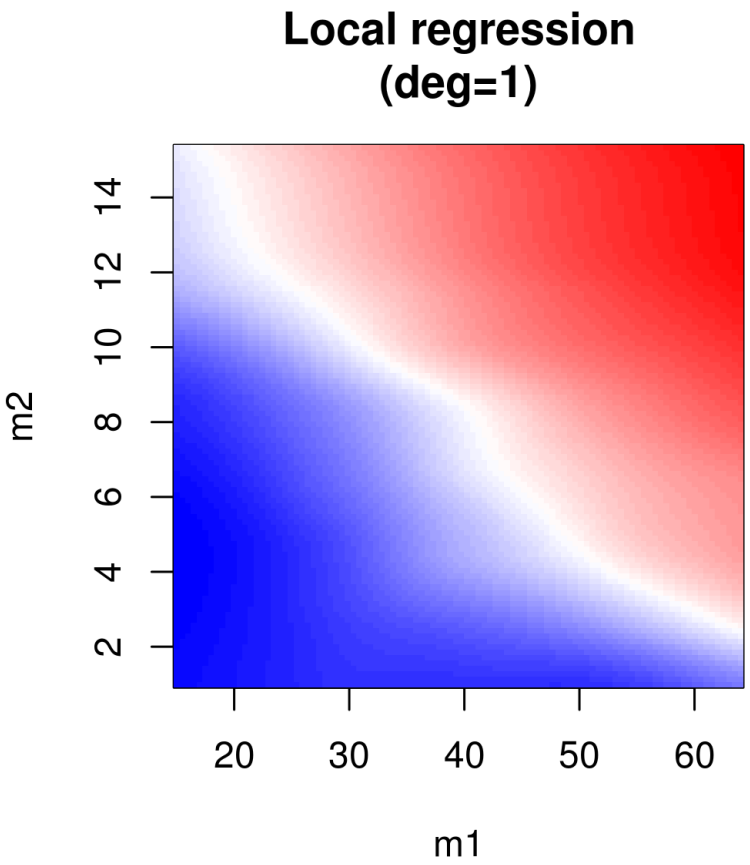
Local regression

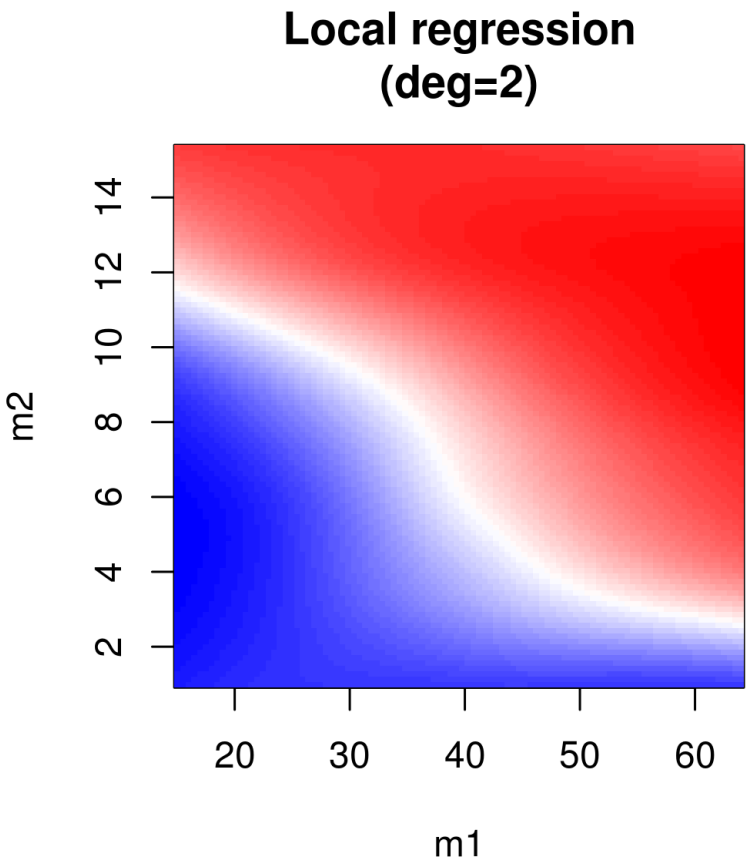
- We have already seen how to extend local regression to the multivariate case, back when we discussed estimating multivariate densities
- All that is required is to define a multivariate kernel:

$$\hat{f}(\mathbf{x}_0) = \frac{1}{n} \sum_i \prod_{j=1}^{p} \frac{1}{h_j} K\left(\frac{x_{ij} - x_{0j}}{h_j} \right)$$

- With the kernel defined, we can now fit a weighted multiple regression model, with elements $x_{ij} - x_{0j}$

Local multiple regression





Thin plate splines

- The multidimensional analog of smoothing splines are called *thin plate splines*
- For two dimensions, we find $f(x_1, x_2)$ that minimizes

$$-\sum_{i=1}^{n} \ell\{y_i, f(x_{1i}, x_{2i})\} + \lambda \int\int \left[\frac{\partial^2 f}{\partial u^2}\right]^2 + 2\left[\frac{\partial^2 f}{\partial u \partial v}\right]^2 + \left[\frac{\partial^2 f}{\partial v^2}\right]^2 du dv$$

- Thin plate splines have fairly complicated basis functions

Scales and isotropy

- An important feature of thin-plate splines is that they are *isotropic*: curvature in all directions is penalized equally
- This makes sense when $f(x_1, x_2)$ is a function of, say, spatial coordinates measured in identical units
- However, if x_1 and x_2 are different quantities measured in units which are not comparable, the isotropy assumption may make little sense and result in a lack of equivariance
- In practice, it is common to rescale variables to have mean 0, variance 1, or so that they can fit on the unit square
- This issue is equally relevant to multiple local regression, where bandwidths $\{h_j\}$ must be chosen

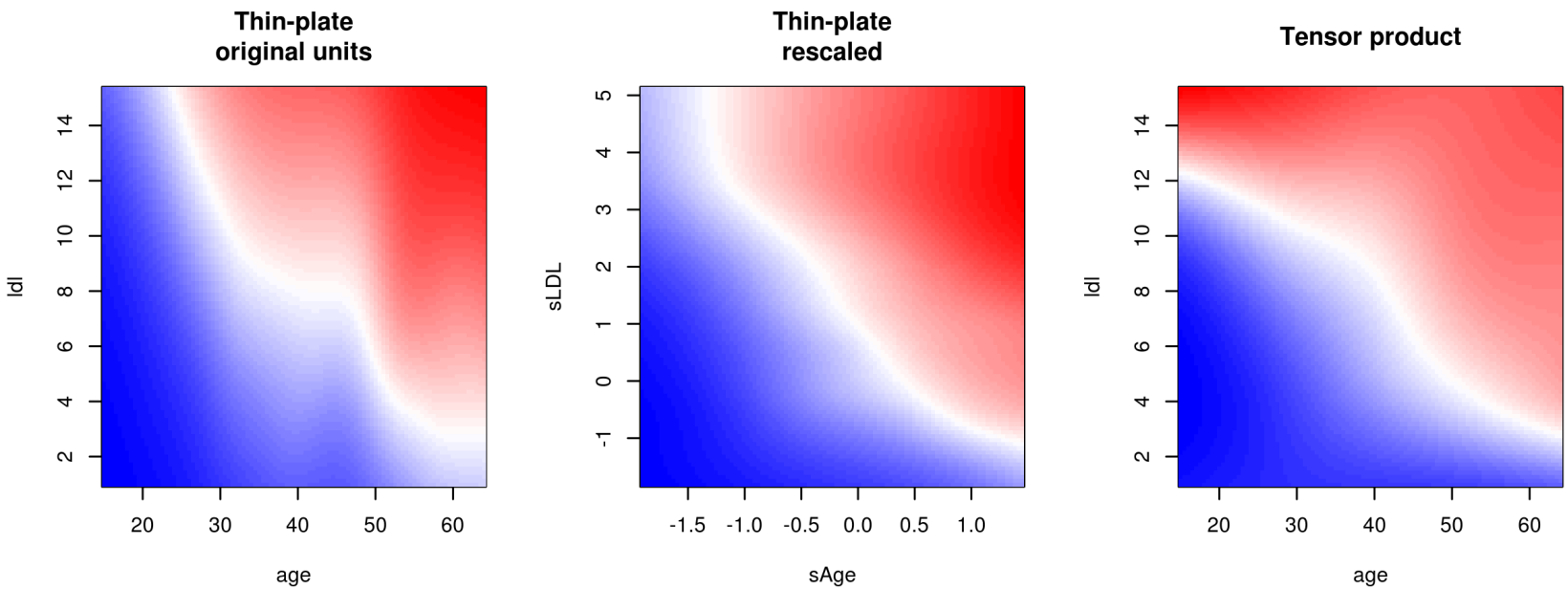
Tensor product splines

- An alternative approach to constructing multidimensional splines is to use a *tensor product basis*
- Suppose we specify a set of basis functions $\{h_{1k}\}$ for x_1 and $\{h_{2k}\}$ for x_2, with $M1$ and $M2$ elements, respectively
- The tensor product basis for the two-dimensional smooth function of x_1 and x_2 is given by

$$g_{jk}(x_1, x_2) = h_{1j}(x_1)h_{2k}(x_2)$$

and has $M1 \times M2$ elements

Multidimensional splines



The curse of dimensionality

- Thin plate splines and tensor product splines can be extended further into higher dimensions, although they become rather computationally intensive as the dimension exceeds 2
- Also, as we saw with kernel density estimation, the curse of dimensionality implies that we need an exponentially increasing amount of data to maintain accuracy as p increases (this applies to both local regression and splines/penalized regression)
- Furthermore, multidimensional smooth functions are harder to visualize and interpret
- Because of this, it is often necessary/desirable to introduce some sort of structure into the model

Structured regression

- Introducing structure will certainly introduce bias if the structure does not accurate describe reality; however, it can result in a dramatic reduction in variance
- Nonparametric multiple regression usually comes down to balancing these goals: introducing enough structure to make the model fit stable, but not so much structure as to bias the fit
- A number of methods have been proposed in the hopes of accomplishing this balance, including structured kernels, varying coefficient models, and projection pursuit regression
- By far the simplest and most common approach, however, is to introduce an additive structure; the resulting model is called an *additive model*

Generalized additive models

- An additive model is of the form

$$\mathbb{E}(y|\mathbf{x}) = \alpha + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)$$

- By introducing a distribution and link function into linear regression, we have generalized linear models (GLMs)
- By introducing a distribution and link function into additive models, we have generalized additive models (GAMs):

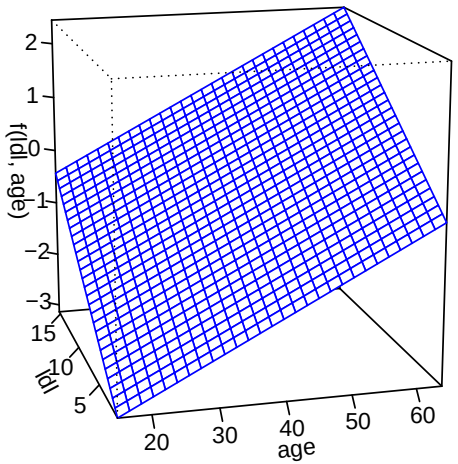
$$g\{\mathbb{E}(y|\mathbf{x})\} = \alpha + \sum_j f_j(x_j)$$

GAMs and the curse of dimensionality

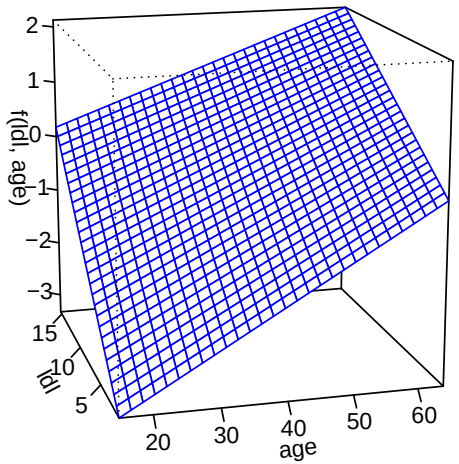
- This additive structure greatly alleviates the curse of dimensionality:
  - From a spline perspective, we need only $\sum_p m_j$ basis functions instead of $\prod_p m_j$ basis functions
  - From a local regression perspective, it is much easier to find points in a one-dimensional neighborhood
- As we will see, additive models are also easy to fit computationally

Restrictions imposed by various models

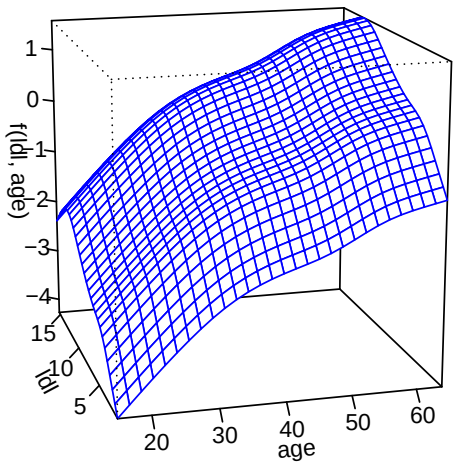
GLM



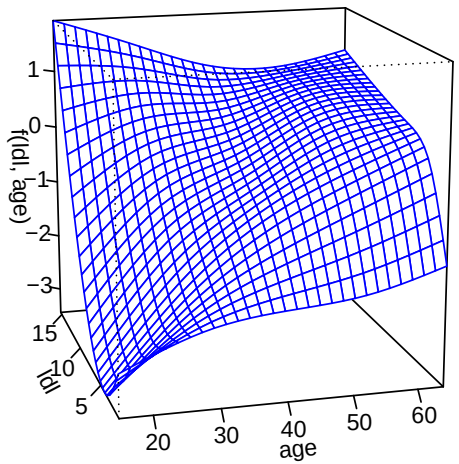
GLM w/ interaction



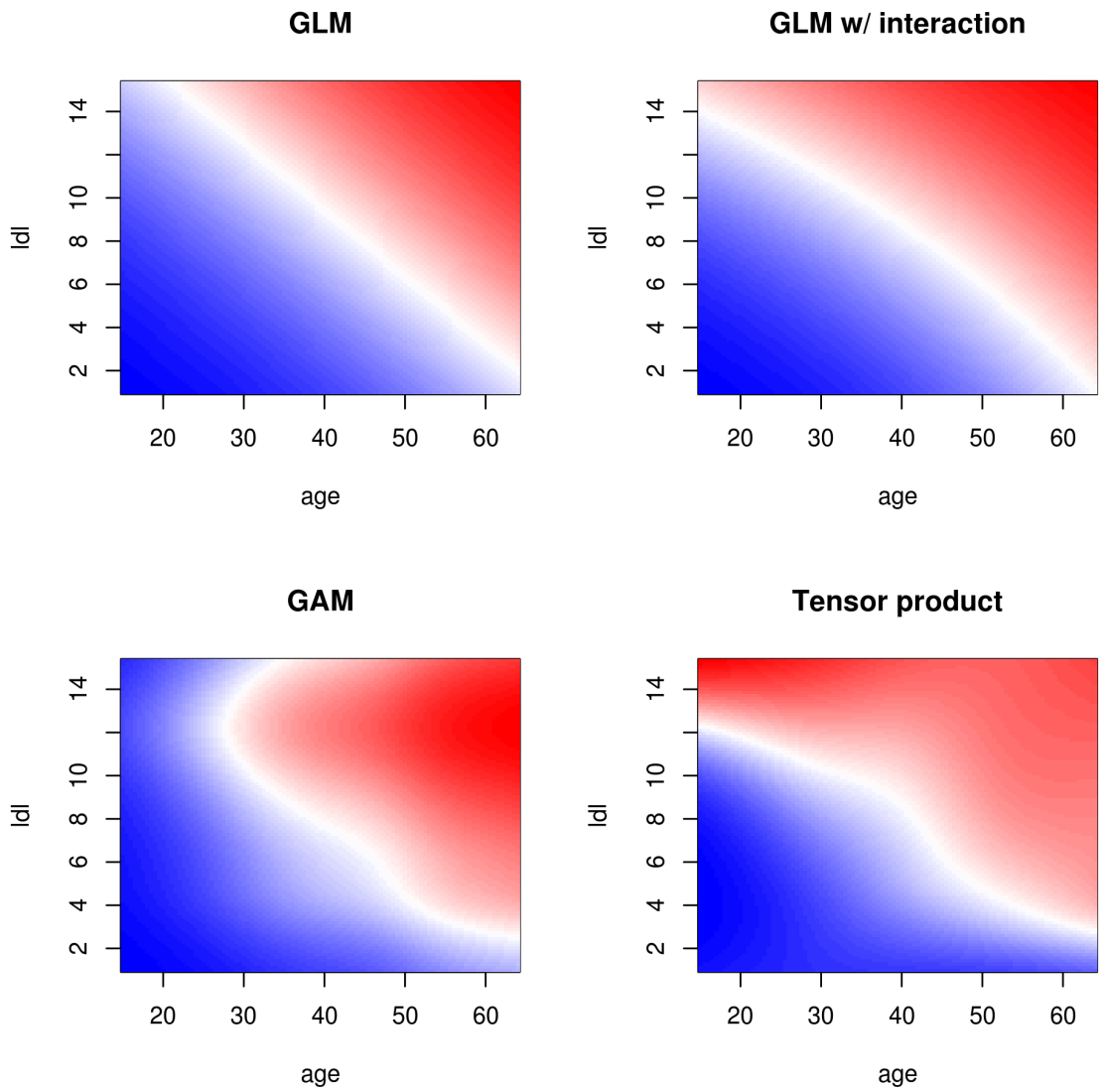
GAM



Tensor product



Restrictions imposed by various models (cont'd)



The gam package

- As we mentioned in a previous lecture, there are two R packages for the implementation of GAMs: mgcv and gam
- The two packages both supply a gam function with a formula interface and are superficially very similar
- However, the implementation which underlies their model fitting is very different, and as a result, they offer different features

Backfitting

The gam package is based on a simple algorithmic approach called *backfitting* for turning any one-dimensional regression smoother into a method for fitting additive models

(1) Initialize: $\hat{\alpha} = \frac{1}{n}\sum_i y_i$, $\hat{f}_j = 0$ for all j

(2) Cycle over j until convergence:

  (a) Compute $\tilde{y}_i = y_i - \hat{\alpha} - \sum_{k \neq j} f_k(x_{ik})$ for all i

  (b) Apply the one-dimensional smoother to $\{x_{ij}, \tilde{y}_i\}$ to obtain $\hat{f}_j$

  (c) Set $\hat{f}_j$ equal to $\hat{f}_j - n^{-1}\sum_i \hat{f}_j(x_{ij})$

Note that we require $\sum_i \hat{f}_j(x_{ij}) = 0$ for all j; otherwise the model is not identifiable

Backfitting (cont'd)

- The modular nature of the backfitting algorithm makes it easy to fit very general models, such as:
  - Models in which some terms are fit via local polynomials and others fit via splines
  - Models that mix parametric and nonparametric terms
  - Models that include 2D smooth functions to model nonparametric interactions of terms
- Computing degrees of freedom is also a simple extension of earlier results: letting $\mathbf{L}_j$ denote the smoother matrix for the jth term, the degrees of freedom of the jth term is $\mathrm{tr}(\mathbf{L}_j) - 1$

Syntax: gam

- So in the gam package, one could submit

  ```
  fit <- gam(y~x1+s(x2)+lo(x3))
  ```

  to fit a model in which x1 is modeled parametrically, x2 is modeled using splines, and x3 is modeled using loess
- As we have seen, we can use the anova method to test nested models:

```
> fit <- gam(chd~s(age)+lo(ldl)+obesity,fam="binomial")
> fit0 <- gam(chd~s(age)+ldl+obesity,fam="binomial")
> anova(fit0,fit)
Model 1: chd ~ s(age) + ldl + obesity
Model 2: chd ~ s(age) + lo(ldl) + obesity
  Resid. Df Resid. Dev     Df Deviance P(>|Chi|)
1    455.00     505.34
2    451.69     504.20 3.3101   1.1364    0.8134
```

The mgcv package

- The syntax of gam in the mgcv package is very similar, although the mgcv package has many more features
- The implementation is based not on backfitting, but rather on the *Lanczos algorithm*, a way of efficiently calculating truncated matrix decompositions
- The implementation is restricted to splines (*i.e.* no mixing of local polynomials and splines)

Selection of λ

- One key advantage of this approach is that it allows for the evaluation of the derivative of AIC with respect to λ_j
- This makes it possible to employ a Newton's method approach to simultaneously fit the model and optimize over the smoothing parameters with respect to AIC
- In practice, this is an attractive advantage over the `gam` package, for which you must specify either λ_j or df_j

Syntax: mgcv

- The basic syntax of gam in the mgcv package is:

```
fit <- gam(chd~te(age,ldl)+s(obesity)+tobacco,
           fam=binomial)
```

  where here, we are allowing a tensor product interaction between age and LDL, an additive nonparametric effect of obesity, and an additive parametric effect of tobacco use on the log odds of coronary heart disease
- One can add arguments to the `te()` and `s()` functions, but the default behavior is to use a natural cubic spline/thin-plate spline basis and to automatically choose the smoothing parameter via optimization of the GCV or AIC objective (which the package calls UBRE)

summary.gam

```
> summary(fit)
Family: binomial
Link function: logit

Formula: chd ~ te(age, ldl) + s(obesity) + tobacco

Parametric coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.10345    0.15277  -7.223 5.09e-13 ***
tobacco      0.07634    0.02556   2.987  0.00282 **

Approximate significance of smooth terms:
               edf Ref.df Chi.sq  p-value
te(age,ldl) 3.619  4.055 44.591 5.66e-09 ***
s(obesity)  1.821  2.333  2.938    0.279

R-sq.(adj) =   0.18   Deviance explained = 16.5%
UBRE score = 0.10897  Scale est. = 1         n = 462
```

anova.gam

More specific hypotheses can be tested via the anova method:

```
> fit0 <- gam(chd ~ te(age,ldl) + obesity + tobacco,
              data=heart, family=binomial)
> anova(fit0, fit, test="Chisq")
Analysis of Deviance Table

Model 1: chd ~ te(age, ldl) + obesity + tobacco
Model 2: chd ~ te(age, ldl) + s(obesity) + tobacco
  Resid. Df Resid. Dev     Df Deviance Pr(>Chi)
1    455.67     501.01
2    454.56     497.46 1.1108   3.5483  0.06947 .
```

anova vs. summary

It is important to note that, while anova and summary agree for linear regression models, in the nonparametric GAM case they are taking different approaches to testing, and do not produce the same results for the same tests:

```
> summary(fit)
...
               edf Ref.df Chi.sq  p-value
s(obesity)  1.821  2.333  2.938    0.279
...
> fit0 <- gam(chd ~ te(age,ldl) + tobacco,
              data=heart, family=binomial)
> anova(fit0, fit, test="Chisq")
Model 1: chd ~ te(age, ldl) + tobacco
Model 2: chd ~ te(age, ldl) + s(obesity) + tobacco
  Resid. Df Resid. Dev     Df Deviance Pr(>Chi)
1    456.98     502.63
2    454.56     497.46 2.4218   5.1721   0.1071
```

anova vs. summary

This is due to three factors:

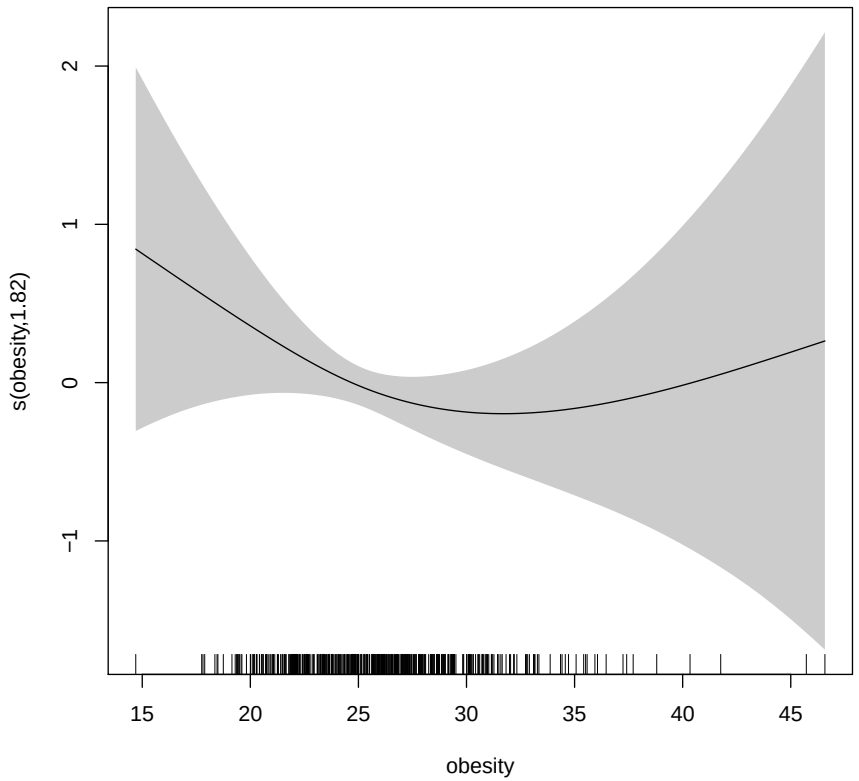
- summary is based on a Wald-type test, while anova is essentially a likelihood ratio test (this issue arises in standard GLMs also)
- When models are refit, the optimal values of λ_j do not stay the same
- The hypothesis tests in summary and anova use different definitions for the effective degrees of freedom; anova uses $\text{tr}(\mathbf{L})$, summary uses $\text{tr}(2\mathbf{L} - \mathbf{L}'\mathbf{L})$

Comments on hypothesis testing

- It should be pointed out once again that these tests are approximate, and should be taken as only a rough guide concerning statistical significance
- This issue is in not unique to nonparametric regression; any time model selection is performed, the resulting p-value are no longer valid and should be taken as only rough indicators of significance
- Keep in mind that hypothesis testing is often not the purpose of the analysis, and that building a model that accurately estimates the relationship between the outcome and explanatory variables may be a more meaningful goal

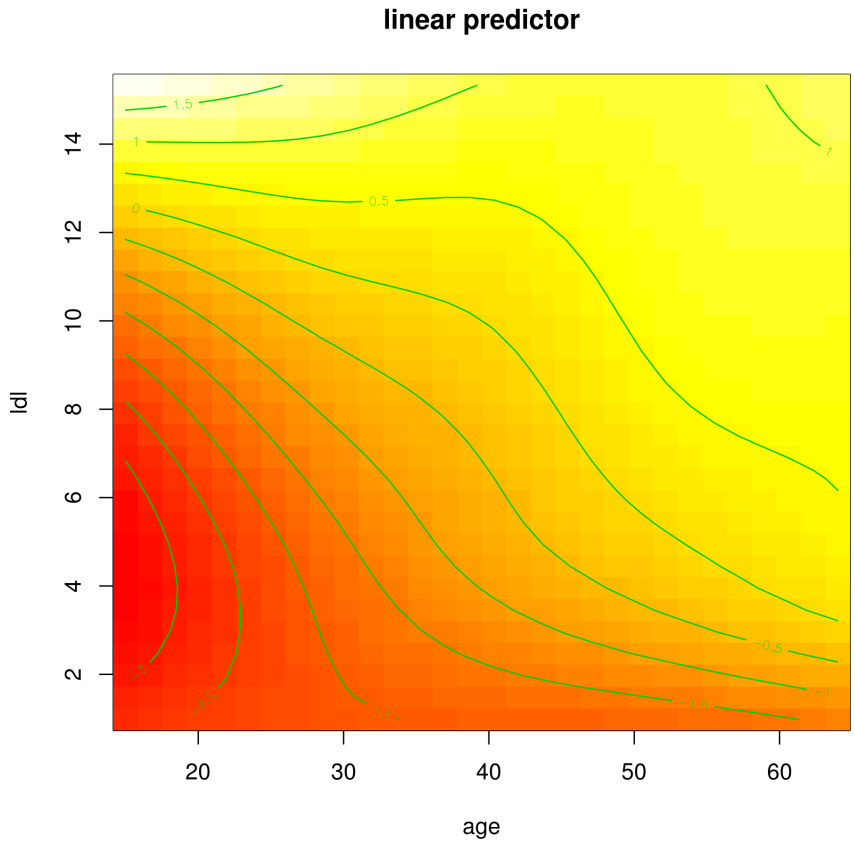
1D plots

```
plot(fit, shade=TRUE)
```



2D plots

```
vis.gam(fit, view=c("age", "ldl"), plot.type="contour")
```



Comments

- Note that, although one may specify a nonparametric form, gam will often return linear or nearly linear fits for some parameters because this is the fit that optimized the AIC criterion
- For example, in the heart study, the age-LDL interaction had 3.6 degrees of freedom, only slightly more flexible than the 3 degrees of freedom arising from a parametric interaction
- This is entirely driven by the data: as an example, I was once working on a project in which I found a meaningful three-way interaction between age, driving distance, and urban/rural location on the probability that an individual would attend an intervention designed to educate women aged 40-64 on living healthier lifestyles

A nonparametric three-way interaction: