6.3 Fitting GAMs with gam() from mgcv package in R

Fitting GAMs with gam() from mgcv package in R

`gam()` fits a generalised additive model (GAM) to data, the term ‘GAM’ being taken to include any quadratically penalised GLM and a variety of other models estimated by a quadratically penalised likelihood type approach (see `family.mgcv`).

The degree of smoothness of model terms is estimated as part of fitting. `gam` can also fit any GLM subject to multiple quadratic penalties (including estimation of the degree of penalisation). Confidence intervals are readily available for any quantity predicted using a fitted model.

Example: Brain Imaging

This section follows an example from Wood (2006).

Data The data are from brain imaging by functional magnetic resonance scanning and were reported in Landau et al. (2003). The `brain` data frame comes with the package `gamair`.

```
library(mgcv)
library(gamair)
data(brain)
head(brain, n=2)
```

```
   X  Y   medFPQ region meanTheta
1 65  9 3.923048     NA  2.838555
2 60 10 0.492985      0  1.218145
```

Here each row of the data frame `brain` corresponds to one voxel. `X` and `Y` give the location of each voxel. `medFPQ` is the brain activity level measurement. We are interested in modeling of `medFPQ` as a function of `X` and `Y`. For these data the main interest lies in cleaning up this particular image.

Preliminary Modelling

Before attempting to fit models, it is worth examining the data itself to look for possible problems: We find that there as some outliers and remove it from our analysis:

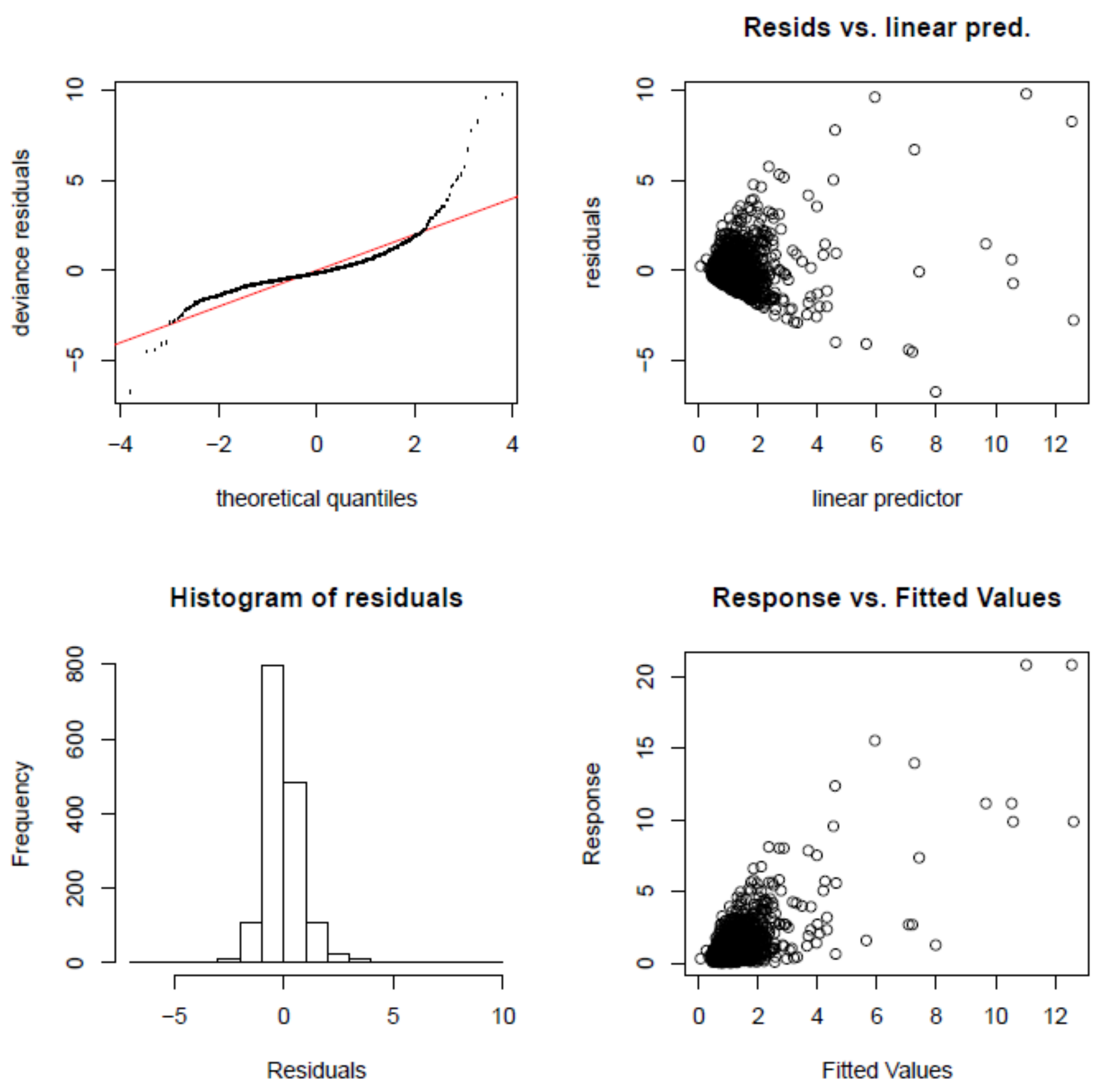
```
library(mgcv)
library(gamair)
data(brain)
attach(brain)
```

```
brain<-brain[medFPQ>5e-3,]
```

Let us now use a Gaussian GAM as follows:

```
library(mgcv)
library(gamair)
data(brain)
attach(brain)

m0<-gam(medFPQ~s(Y,X,k=250))
gam.check(m0)
```

```
Method: GCV Optimizer: magic
Smoothing parameter selection converged after 5 iterations.
The RMS GCV score gradient at convergence was 0.0001210641 .
The Hessian was positive definite.
Model rank = 250 / 250

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k

         k' edf k-index p-value
s(Y,X) 249 169    1.05    0.97
```

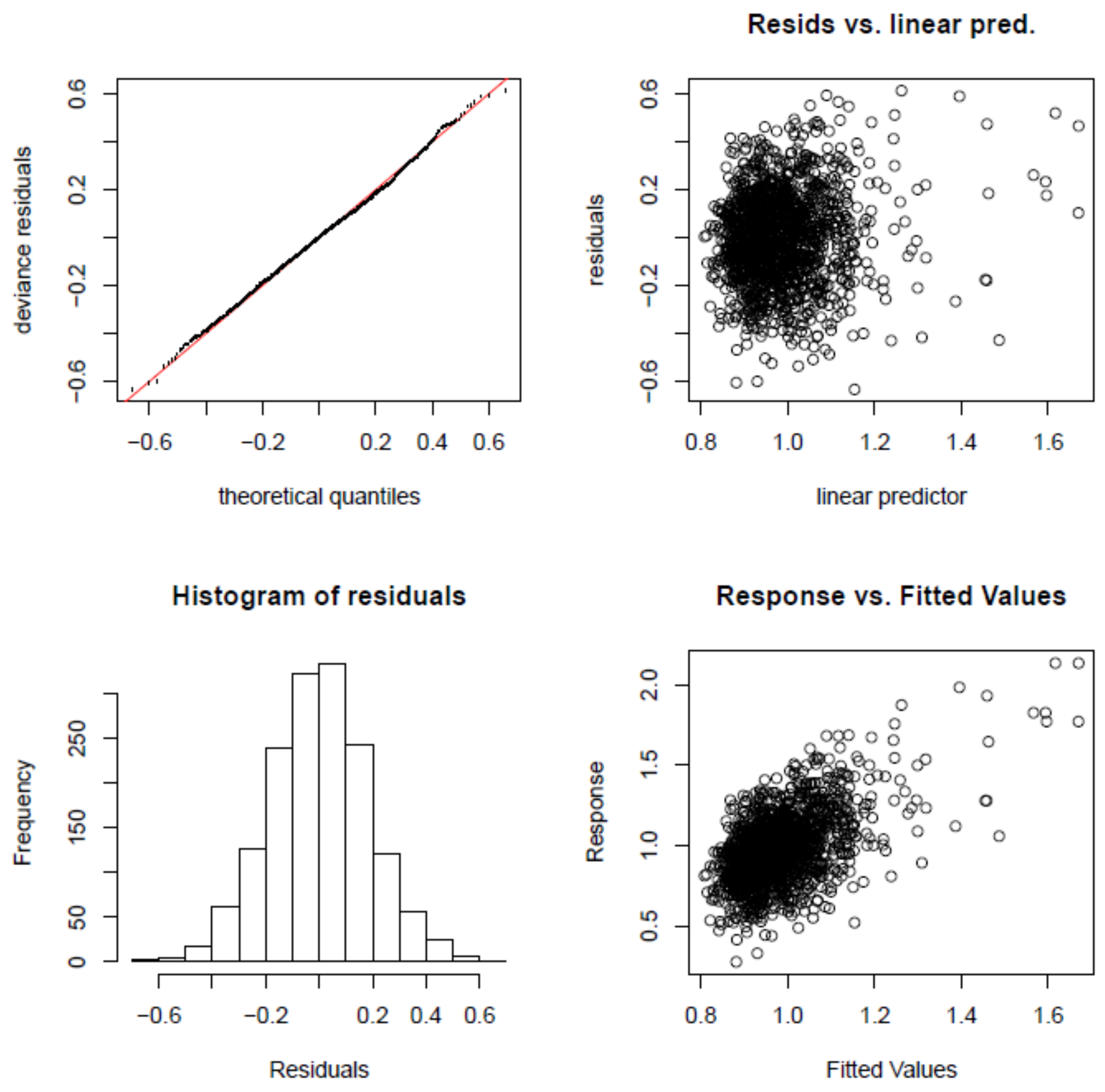


`gam.check` is a routine that produces some basic residual plots. There are clear problems with the constant variance assumption: variance is increasing. Moreover, the fairly skewed nature of the response data, `medFPQ`, suggests that some transformation may be required if a Gaussian error model is to be used. Let us now consider two cases, one with a transformed response and one with Gamma distribution for the response variable.

```
library(mgcv)
library(gamair)
data(brain)
attach(brain)
brain<-brain[medFPQ>5e-3,]
m1<-gam(medFPQ^.25~s(Y,X,k=250),data=brain)
```

```
gam.check(m1)
```

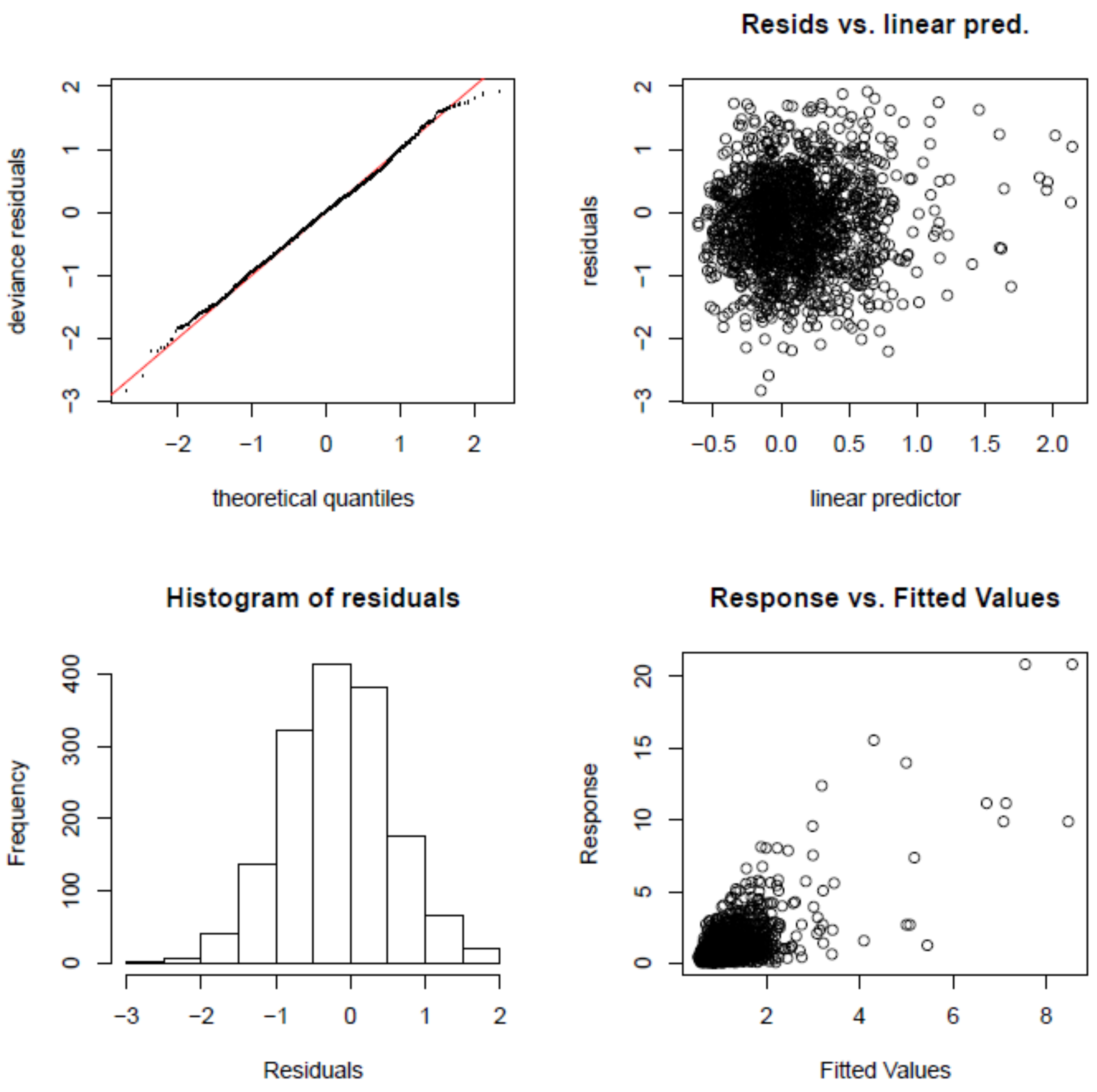
```
Method: GCV Optimizer: magic
Smoothing parameter selection converged after 4 iterations.
The RMS GCV score gradient at convergence was 1.689325e-08 .
The Hessian was positive definite.
Model rank = 250 / 250

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k

         k' edf k-index p-value
s(Y,X) 249 108    0.97    0.05 *
```



and

```
library(mgcv)
library(gamair)
data(brain)
attach(brain)
brain<-brain[medFPQ>5e-3,]
m2<-gam(medFPQ~s(Y,X,k=250),data=brain,family=Gamma(link=log))
gam.check(m2)
```

```
Method: GCV Optimizer: outer newton
full convergence after 2 iterations.
Gradient range [-5.778088e-08,-5.778088e-08]
(score 0.6544238 & scale 0.5493784).
Hessian positive definite, eigenvalue range [0.007412456,0.007412456].
Model rank = 250 / 250

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k

          k'   edf k-index p-value
s(Y,X) 249.0  94.8    0.91    0.05 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



The upper left normal QQ plot is very close to a straight line, suggesting that the distributional assumption is reasonable. The upper right plot suggests that variance is approximately constant as the mean increases. The histogram of residuals is approximately consistent with normality. The lower right plot of response against fitted values shows a positive linear relationship with a good deal of scatter: nothing problematic. The diagnostic plots for `m2` are very similar.

Note also if the response scale is the scale of prime interest, then the Gamma model is to be preferred to the model based on the normality of the transformed data, since:

```
library(mgcv)
library(gamair)
data(brain)
attach(brain)
```

```
brain<-brain[medFPQ>5e-3,]
m1<-gam(medFPQ^.25~s(Y,X,k=250),data=brain)
m2<-gam(medFPQ~s(Y,X,k=250),data=brain,family=Gamma(link=log))

mean(fitted(m1)^4)
mean(fitted(m2))
mean(brain$medFPQ)
```

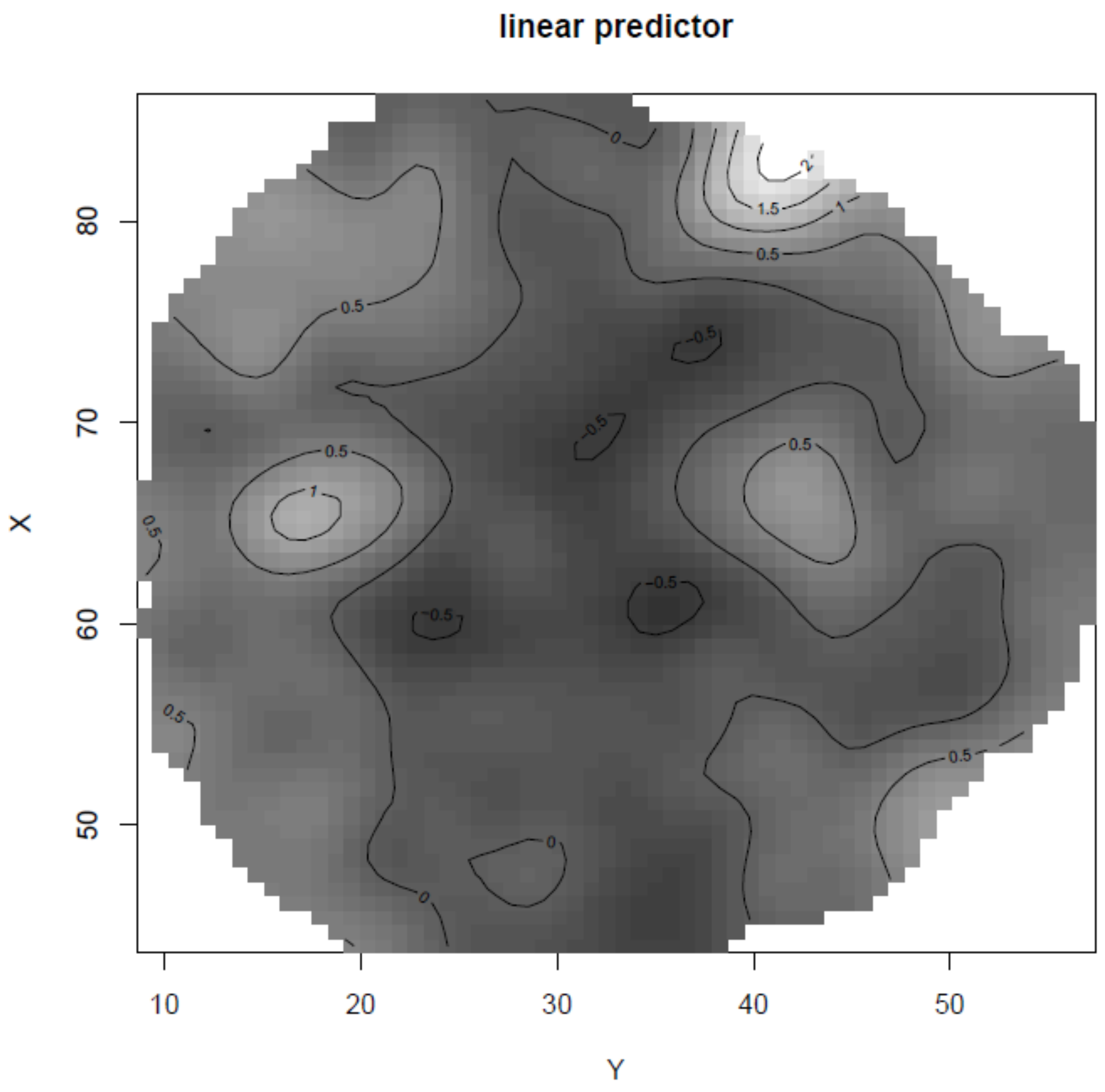
```
[1] 0.9957281
[1] 1.200242
[1] 1.250302
```

This suggests that the model `m1` is biased downwards on the response scale itself. The log-Gamma model is approximately unbiased on the response scale itself.

Let us, therefore, examine the log-Gamma model a little further:

```
library(mgcv)
library(gamair)
data(brain)
attach(brain)
brain<-brain[medFPQ>5e-3,]
m2<-gam(medFPQ~s(Y,X,k=250),data=brain,family=Gamma(link=log))
m2

vis.gam(m2, plot.type="contour", too.far=0.03,color="gray", n.grid=60,zlim=c(-1,2))
```

```
Family: Gamma
Link function: log

Formula:
medFPQ ~ s(Y, X, k = 250)

Estimated degrees of freedom:
94.8  total = 95.79

GCV score: 0.6148743
```



Would an additive structure be better?

Given a large number of degrees of freedom in `m2`, the question arises of whether a different, simpler, model structure might achieve a more parsimonious fit. The obvious candidate is an additive model

$$\log(E[\texttt{medFPQ}_\texttt{i}]) = f_1(\texttt{Y}_\texttt{i}) + f_2(\texttt{X}_\texttt{i}), \quad \texttt{medFPQ}_\texttt{i} \sim \text{Gamma}.$$

```
library(mgcv)
library(gamair)
data(brain)
attach(brain)
brain<-brain[medFPQ>5e-3,]

m3<-gam(medFPQ~s(Y,k=30)+s(X,k=30),data=brain,family=Gamma(link=log))
```

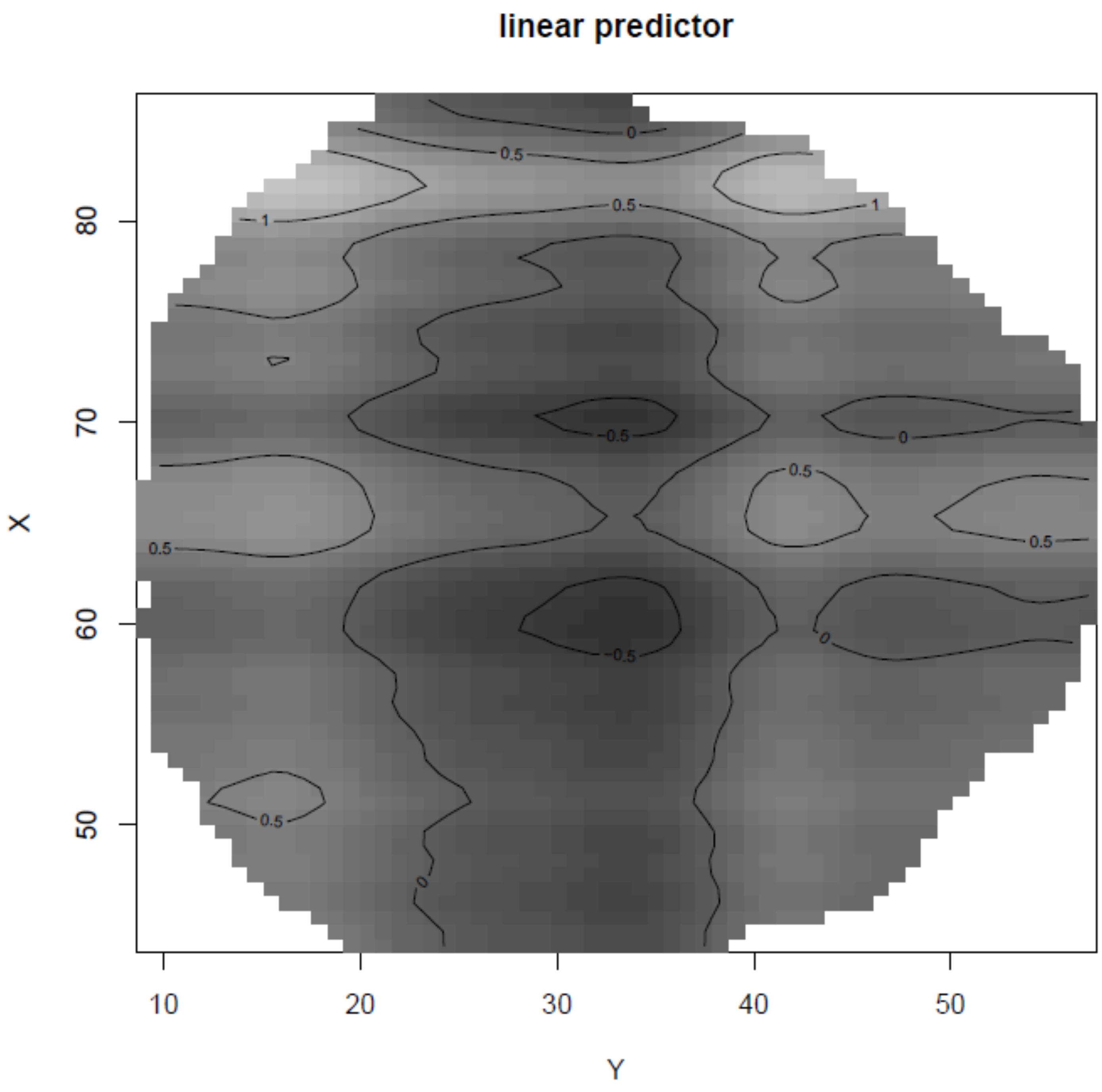
```
m3
```

```
Family: Gamma
Link function: log

Formula:
medFPQ ~ s(Y, k = 30) + s(X, k = 30)

Estimated degrees of freedom:
9.58 20.20  total = 30.77

GCV score: 0.6453502
```

Clearly the GCV is higher for the additive model, suggesting that it is not an improvement.

Plotting the output of the fit from the additive model produces horizontal and vertical stripes that have no real support from the data.

```
library(mgcv)
library(gamair)
data(brain)
attach(brain)
brain<-brain[medFPQ>5e-3,]
m3<-gam(medFPQ~s(Y,k=30)+s(X,k=30),data=brain,family=Gamma(link=log))
vis.gam(m3, plot.type="contour", too.far=0.03,color="gray", n.grid=60,zlim=c(-1,2))
```



Comparison of the AIC criterion for models **m3** and **m2** suggests that model **m2** is a better choice in this case.

```
library(mgcv)
library(gamair)
data(brain)
attach(brain)
brain<-brain[medFPQ>5e-3,]

m2<-gam(medFPQ~s(Y,X,k=250),data=brain,family=Gamma(link=log))
m3<-gam(medFPQ~s(Y,k=30)+s(X,k=30),data=brain,family=Gamma(link=log))
AIC(m3,m2)
```

```
         df      AIC
m3 31.77467 3409.491
```

```
m2 96.79151 3291.364
```

Activity in R: Choice of smoothing basis in mgcv

There is a variety of smoothing basis built into **mgcv**. The basis is specified within the model formula, for example: $\mathtt{s}(, \mathtt{bs} = \texttt{"cr"})$. If not specified default $\mathtt{bs} = \texttt{"tp"}$ is used.

Familiarise yourself with the help file of the **smooth.terms** in the **mgcv** package.

```
library(mgcv)
help(smooth.terms)
```

Match the smoothing basis in R with its description:

(1) **tp** (2) **ts** (3) **cr** (4) **cs** (5) **cc** (6) **ps**

(a) TPRS with shrinkage.
(b) cyclic CRS. As CRS but start point the same as an endpoint.
(c) cubic regression spline (CRS). Produces directly interpretable parameters. Can only smooth with respect to one covariate.
(d) thin plate regression splines (TPRS). Can smooth any number of covariates.
(e) CRS with shrinkage.
(f) P-splines.