# Smoothing and regression splines

Biel Caballero Vergés, Svenja Menzenbach and Kleber Enrique Reyes Illescas

2023-12-13

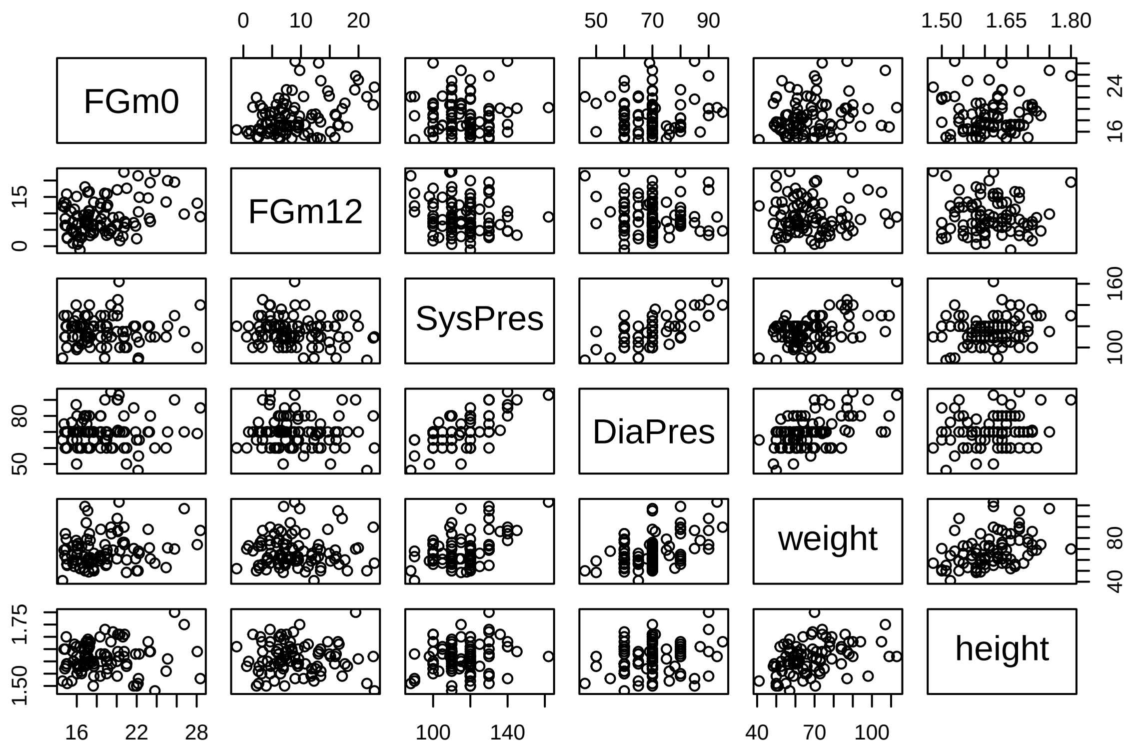
## Hirsutism dataset

### GAMs for hirsutism data

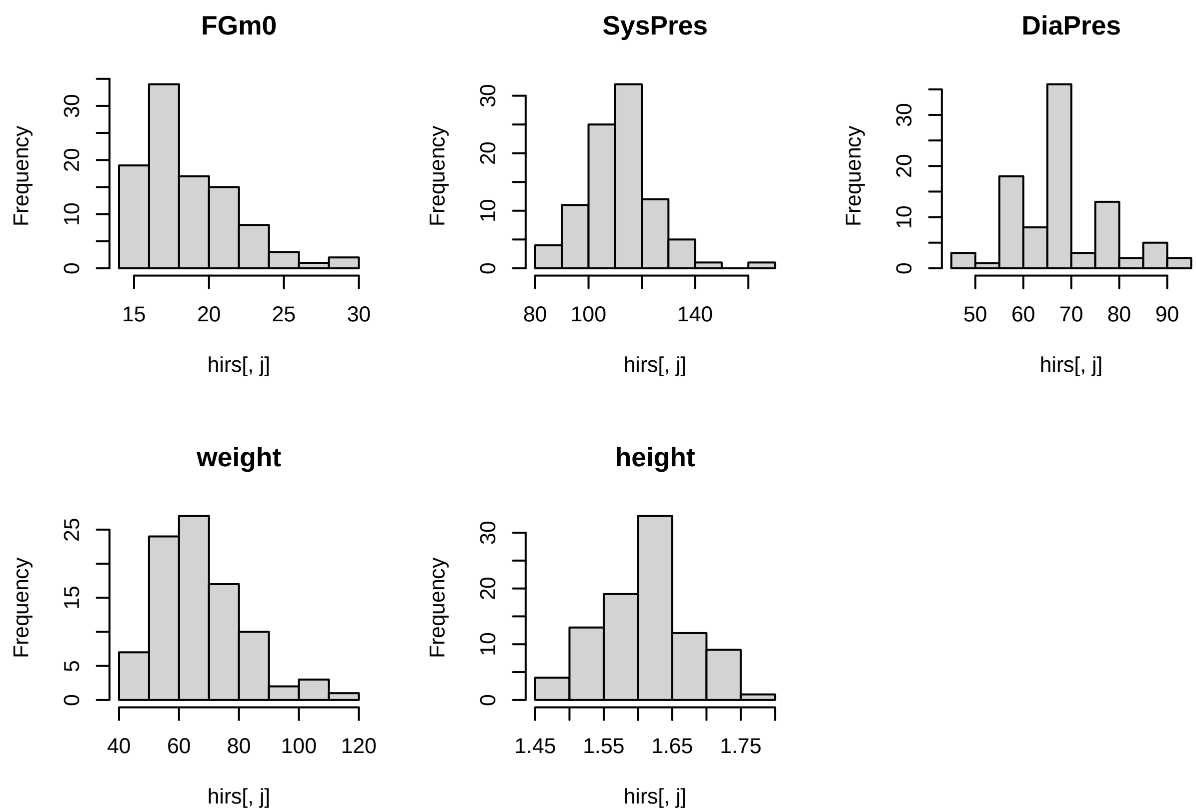
```
hirs <- read.table("hirsutism.dat",header=T, sep="\t",fill=TRUE)

#summary(hirs)
attach(hirs)
```

```
plot(hirs[,-c(1,3,4)])
```



```r
old.par<-par(mfrow=c(2,3))
for (j in c(2,6,7,8,9)) hist(hirs[,j],main=names(hirs)[j])
par(old.par)
```




```r
apply(hirs[,-c(1,3,4,5)],2,sd, na.rm = T) # sd: standard deviation
```

```
##        FGm0     SysPres     DiaPres      weight      height
##  3.10757666 12.99193889  9.59503586 14.72718561  0.06228749
```

```r
apply(hirs[,-c(1,3,4,5)],2,function(x){diff(range(x, na.rm = T))})
```

```
##       FGm0    SysPres    DiaPres     weight     height
## 13.7883505 74.0000000 49.0000000 72.0000000  0.3199999
```

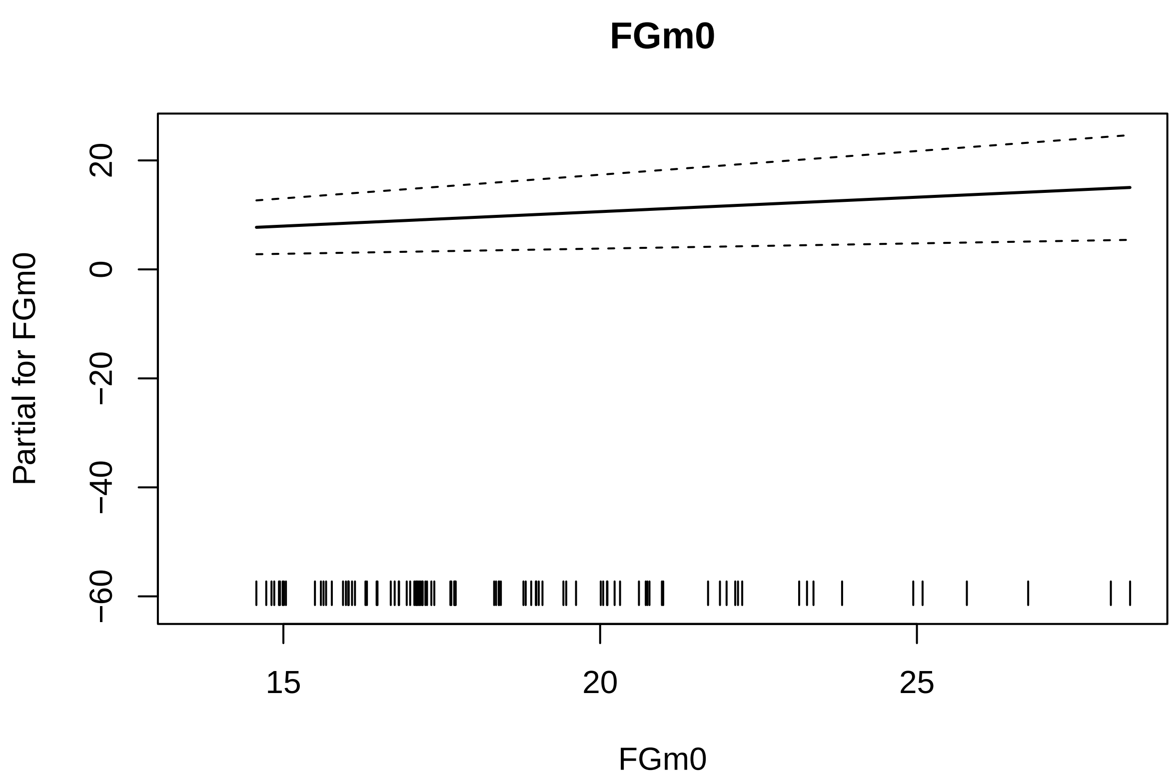
There isn't a clear way to make subgroups (based on dispersion). It's important to note that SysPres and DiaPres appear highly correlated (both variables are related with blood pressure).

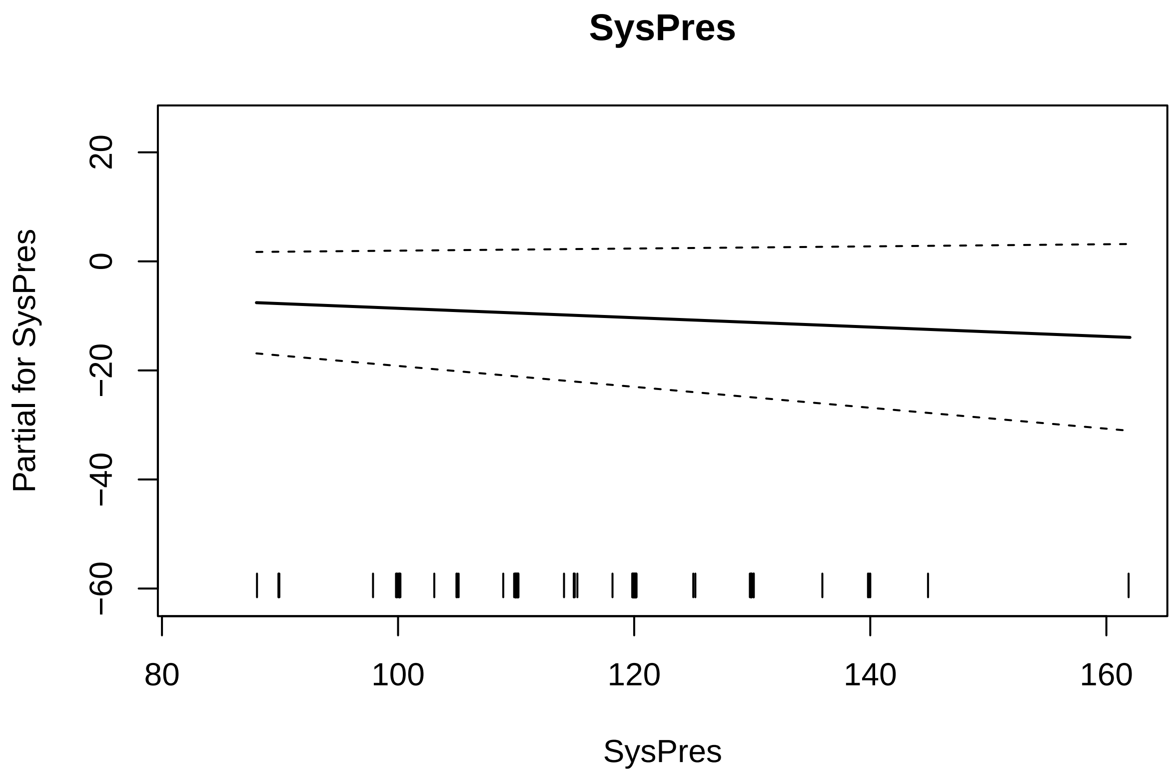
At first we will build a linear model including all variables.

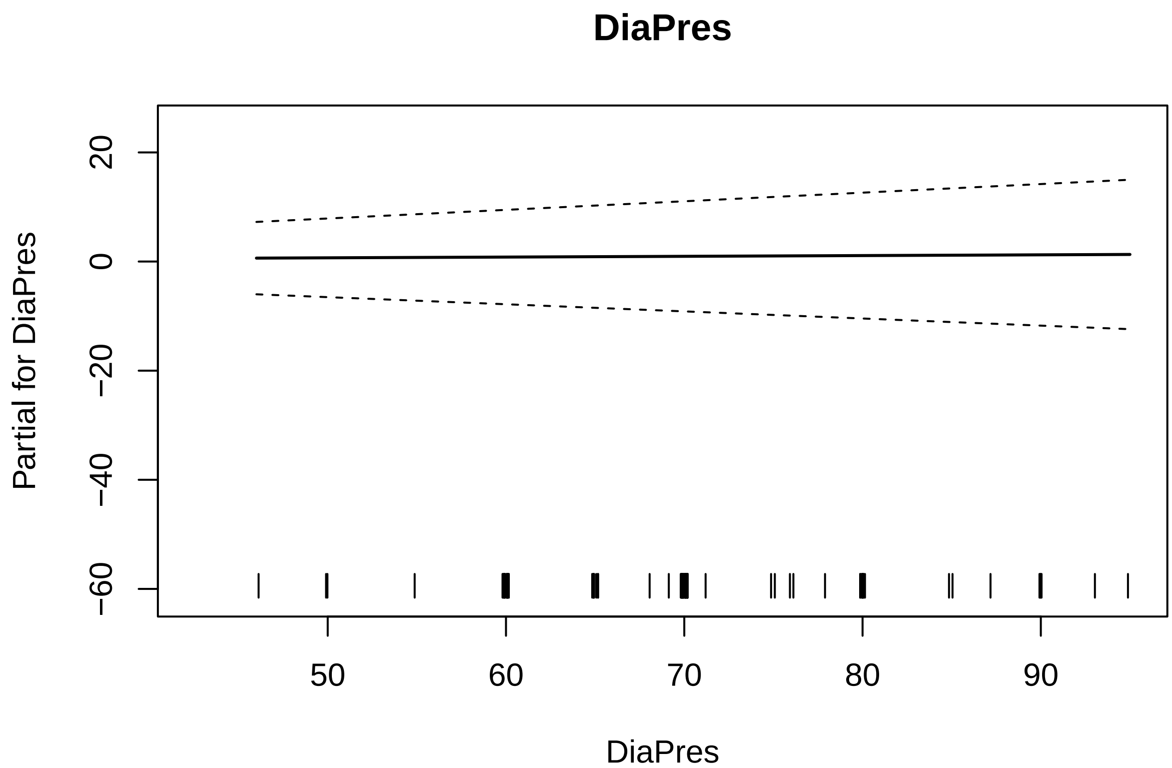
```r
# multiple linear regression
gam_0.1 <- gam(FGm12 ~ FGm0 + SysPres + DiaPres + height + weight + Treatment)
summary(gam_0.1)
```

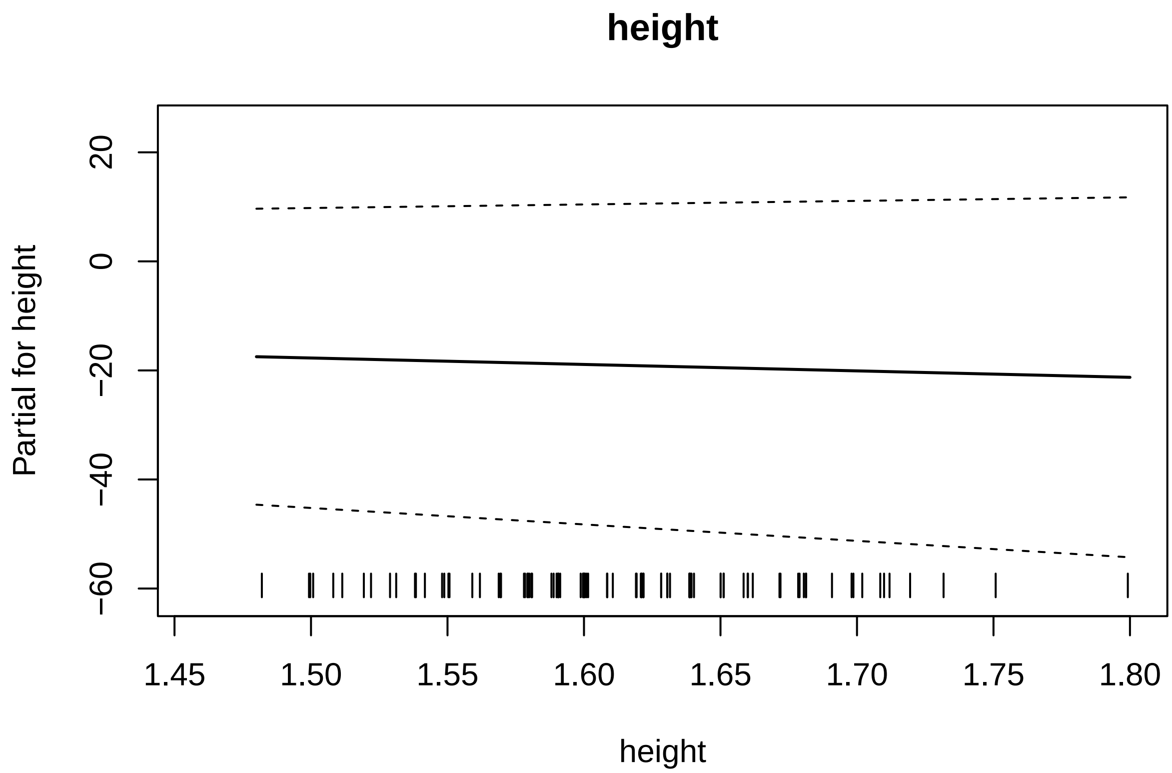
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## FGm12 ~ FGm0 + SysPres + DiaPres + height + weight + Treatment
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.26663   14.82851   1.771  0.08013 .
## FGm0          0.52999    0.16948   3.127  0.00243 **
## SysPres      -0.08608    0.05285  -1.629  0.10712
## DiaPres       0.01364    0.07207   0.189  0.85033
## height      -11.81451    9.16867  -1.289  0.20108
## weight        0.04081    0.04475   0.912  0.36446
## Treatment    -1.16799    0.47107  -2.479  0.01516 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.134   Deviance explained = 19.2%
## GCV = 25.604  Scale est. = 23.634    n = 91
```

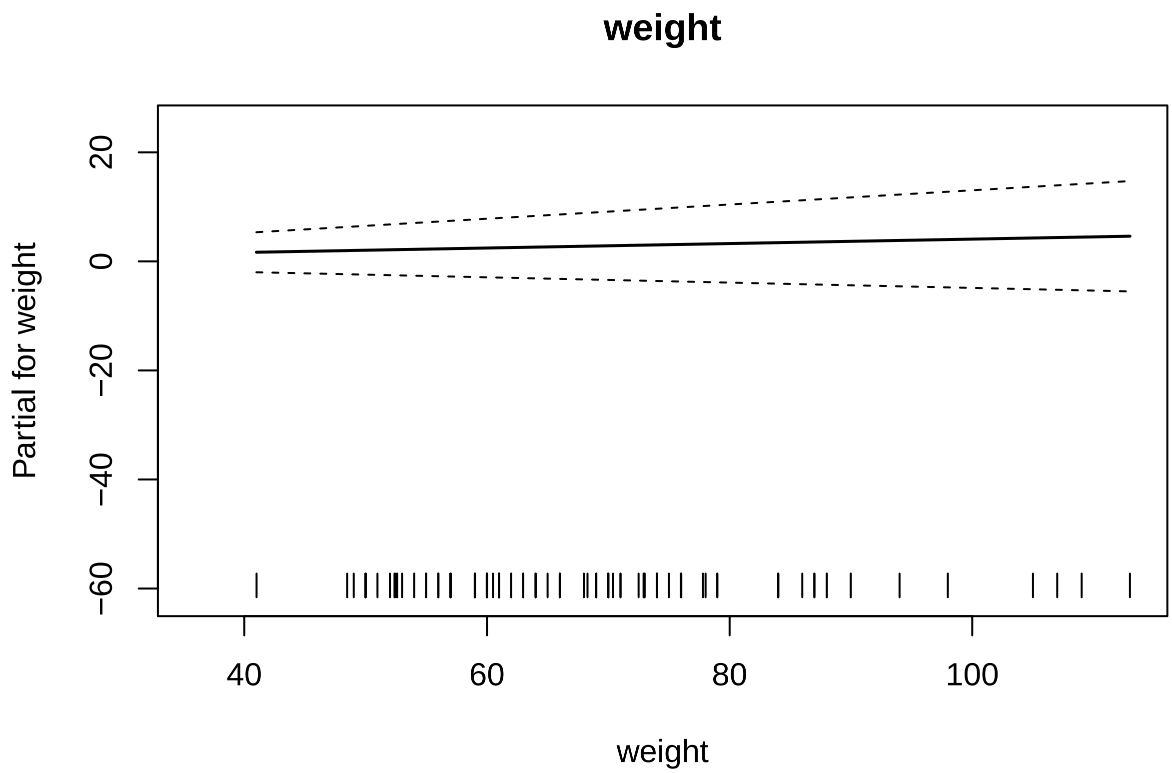
```r
plot(gam_0.1, all.terms=TRUE)
```

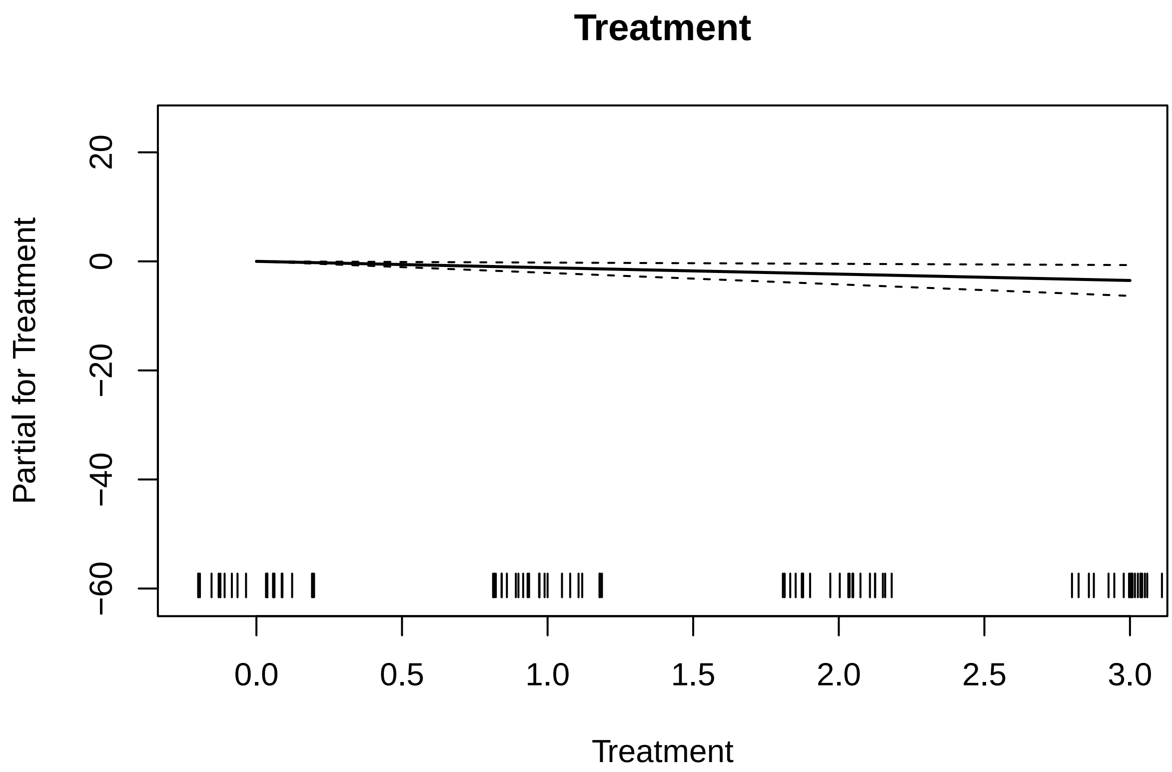



SysPres
Partial for SysPres
20
0
−20
−40
−60
80
100
120
140
160
SysPres


DiaPres
Partial for DiaPres
20
0
−20
−40
−60
50
60
70
80
90
DiaPres


height
Partial for height
20
0
−20
−40
−60
1.45
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height


weight
Partial for weight
20
0
−20
−40
−60
40
60
80
100
weight




All variables show a nearly constant partial which concludes to a bad model as no variable seems to have an effect on the outcome.

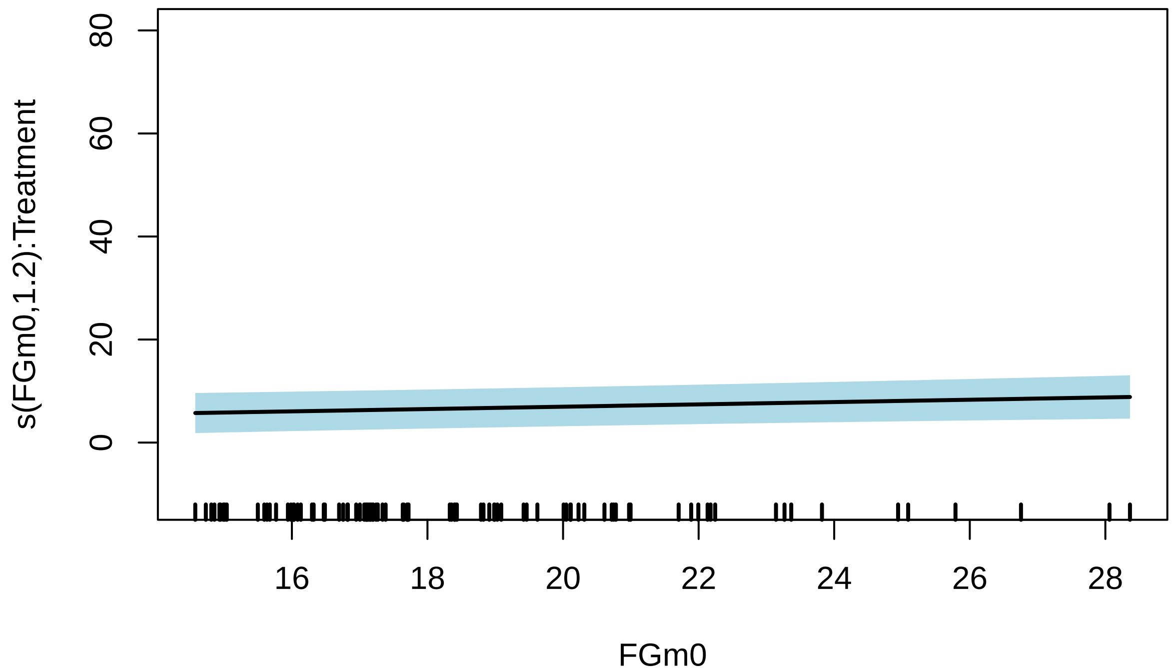
Next we will try a model with only smoothing functions and using Treatment as a factor.

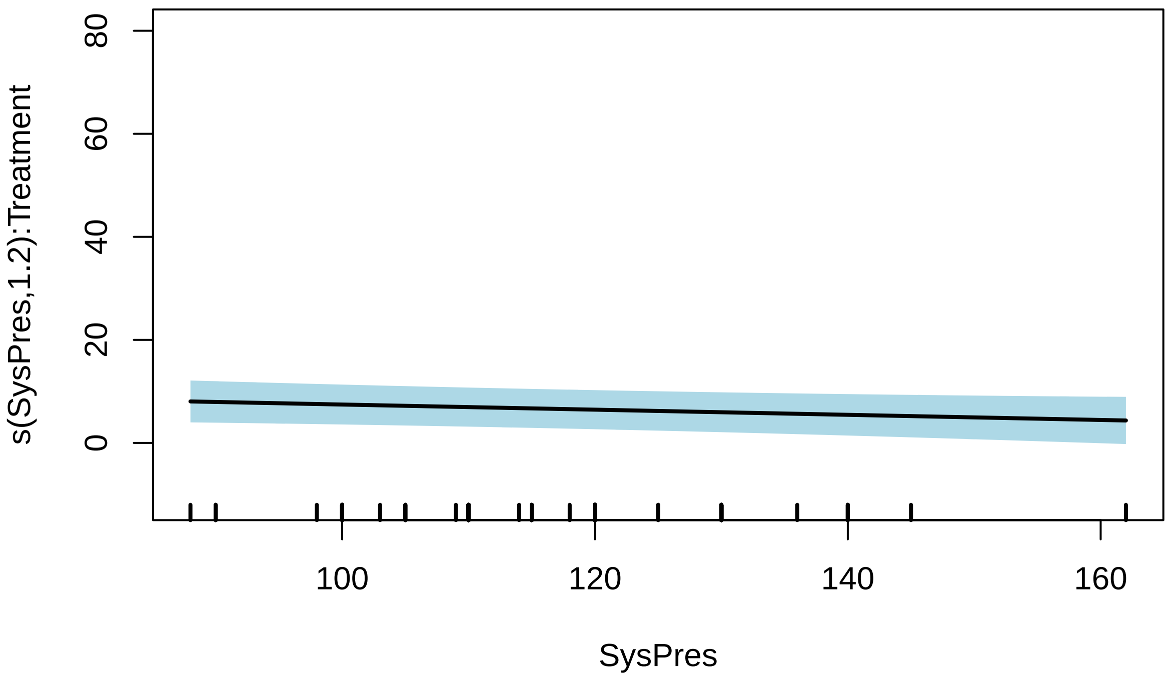
```
# generative additive model
gam_1.1 <- gam(FGm12 ~ s(FGm0, by=Treatment) + s(SysPres, by=Treatment) + s(DiaPres, by=Treatment) + s(
summary(gam_1.1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## FGm12 ~ s(FGm0, by = Treatment) + s(SysPres, by = Treatment) +
##     s(DiaPres, by = Treatment) + s(height, by = Treatment) +
##     s(weight, by = Treatment)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.3098     0.7803   14.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf Ref.df     F p-value
## s(FGm0):Treatment    1.200  1.200 8.478  0.0350 *
```

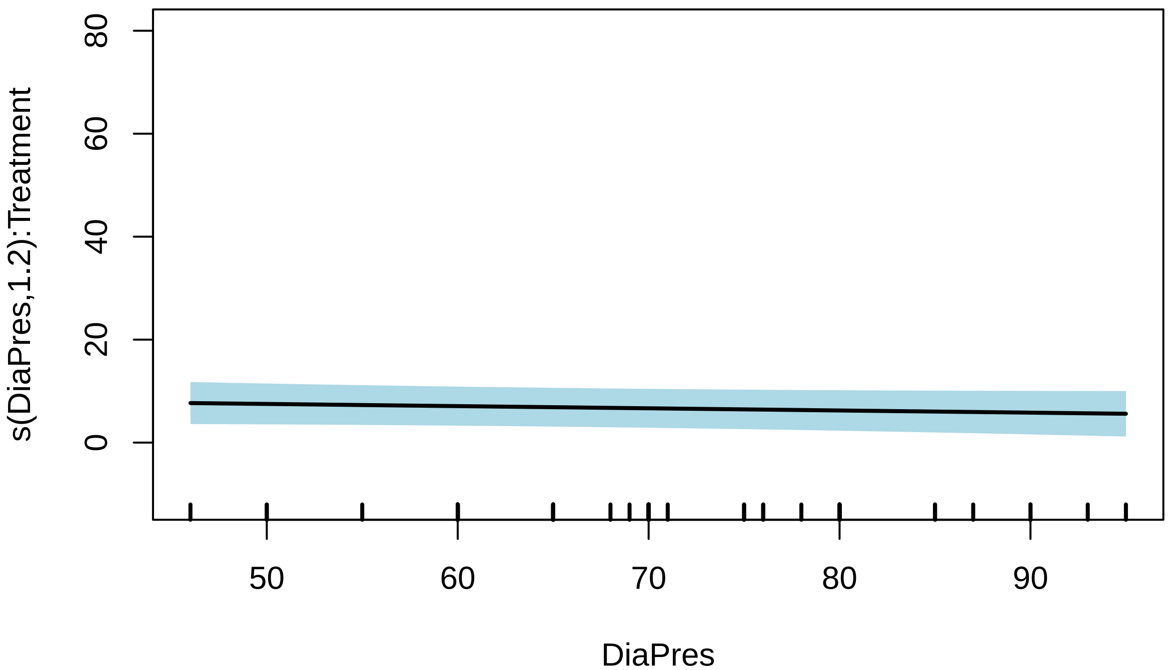
```
## s(SysPres):Treatment 1.200  1.200 3.464  0.0318 *
## s(DiaPres):Treatment 1.200  1.200 6.148  0.0244 *
## s(height):Treatment  7.890  8.685 2.577  0.0172 *
## s(weight):Treatment  5.221  6.289 1.806  0.1086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 47/51
## R-sq.(adj) =  0.345   Deviance explained = 46.7%
## GCV = 22.208  Scale est. = 17.885    n = 91
```

```r
plot(gam_1.1,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= co
```

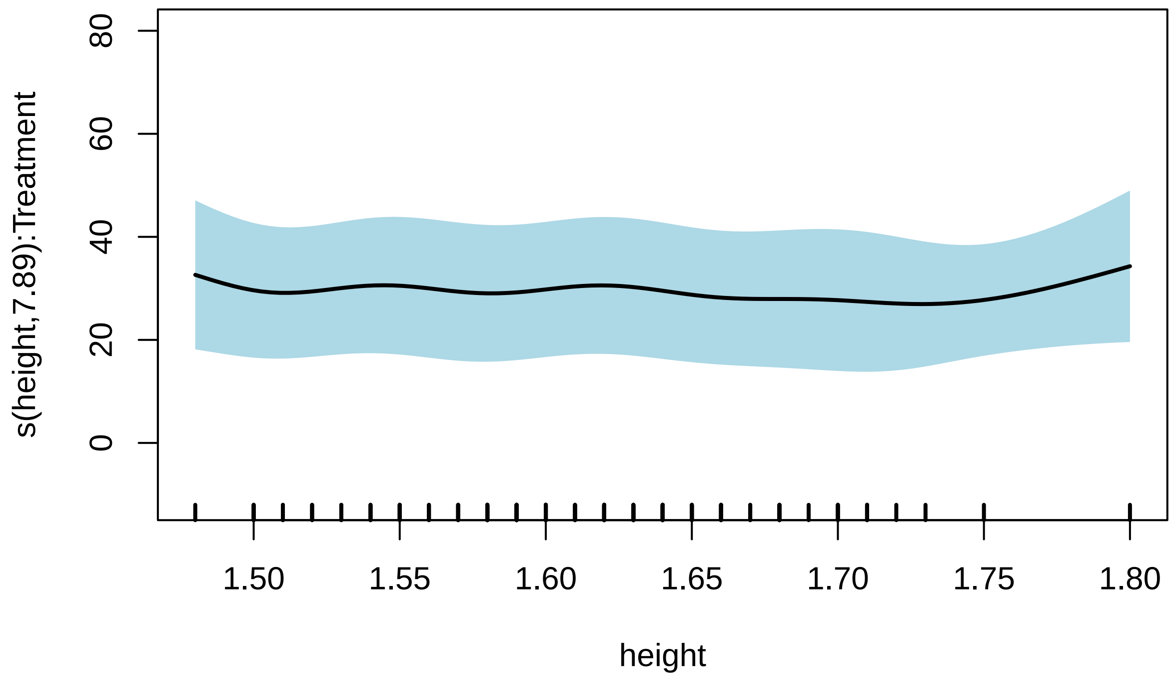


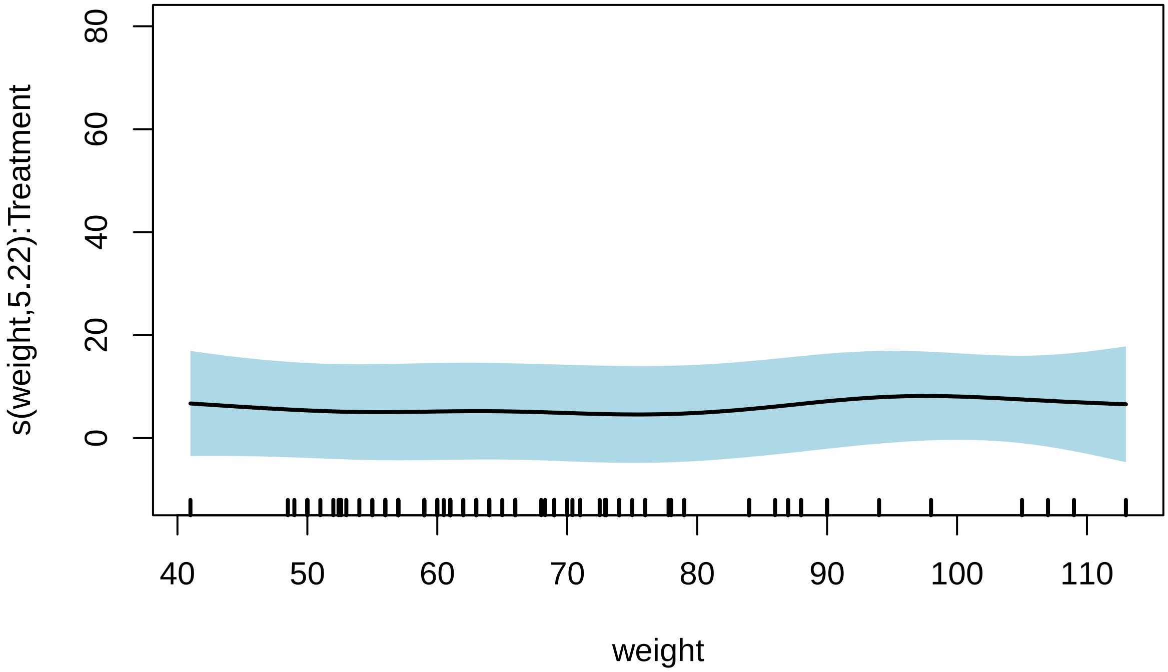


s(SysPres,1.2):Treatment
0
20
40
60
80
100
120
140
160
SysPres

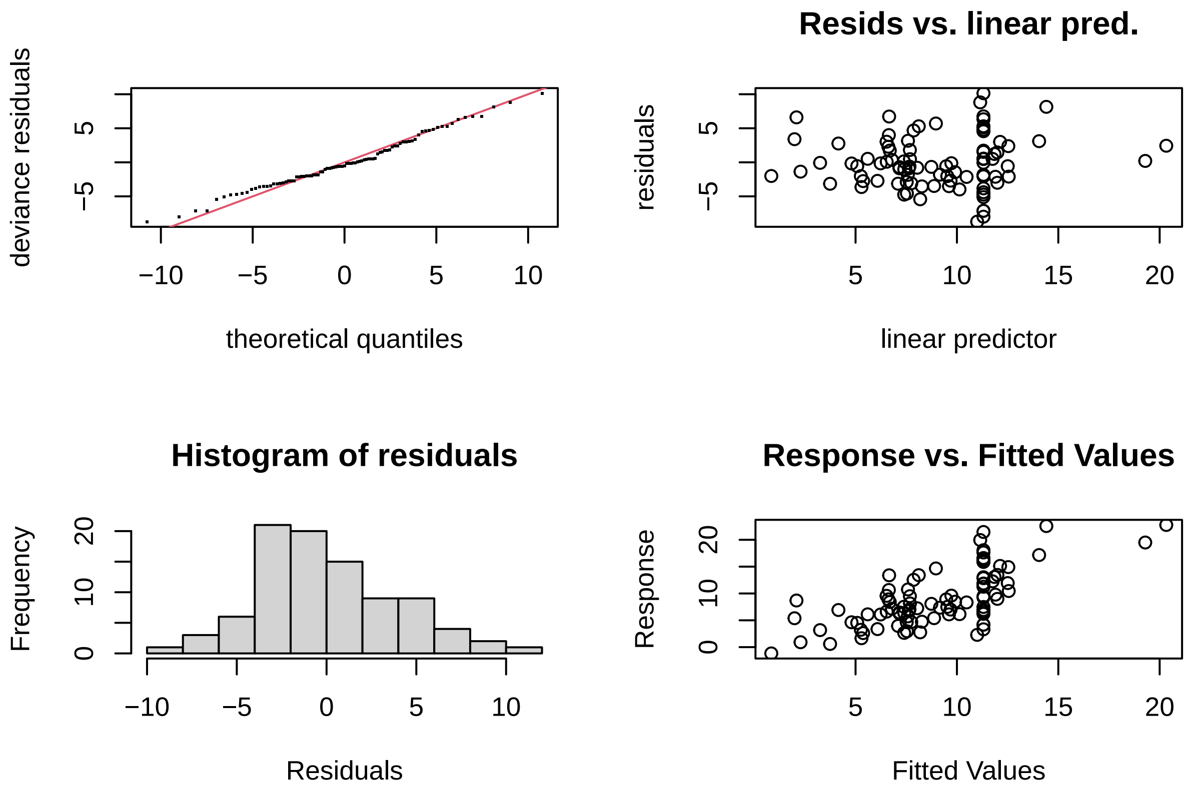


s(DiaPres,1.2):Treatment
0
20
40
60
80
50
60
70
80
90
DiaPres


s(height,7.89):Treatment
0
20
40
60
80
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height




```
gam.check(gam_1.1)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 21 iterations.
## The RMS GCV score gradient at convergence was 5.677106e-07 .
## The Hessian was positive definite.
## Model rank =  47 / 51
##
## 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(FGm0):Treatment    10.00  1.20    1.12    0.84
## s(SysPres):Treatment 10.00  1.20    1.03    0.59
## s(DiaPres):Treatment 10.00  1.20    1.03    0.52
## s(height):Treatment  10.00  7.89    0.94    0.20
## s(weight):Treatment  10.00  5.22    1.05    0.64
```

The partial plots are still quite constant. Hence we didn't find a good model yet. However, the deviance explained, giving insight about the quality of the fit, is more than twice as high as in the previous models.

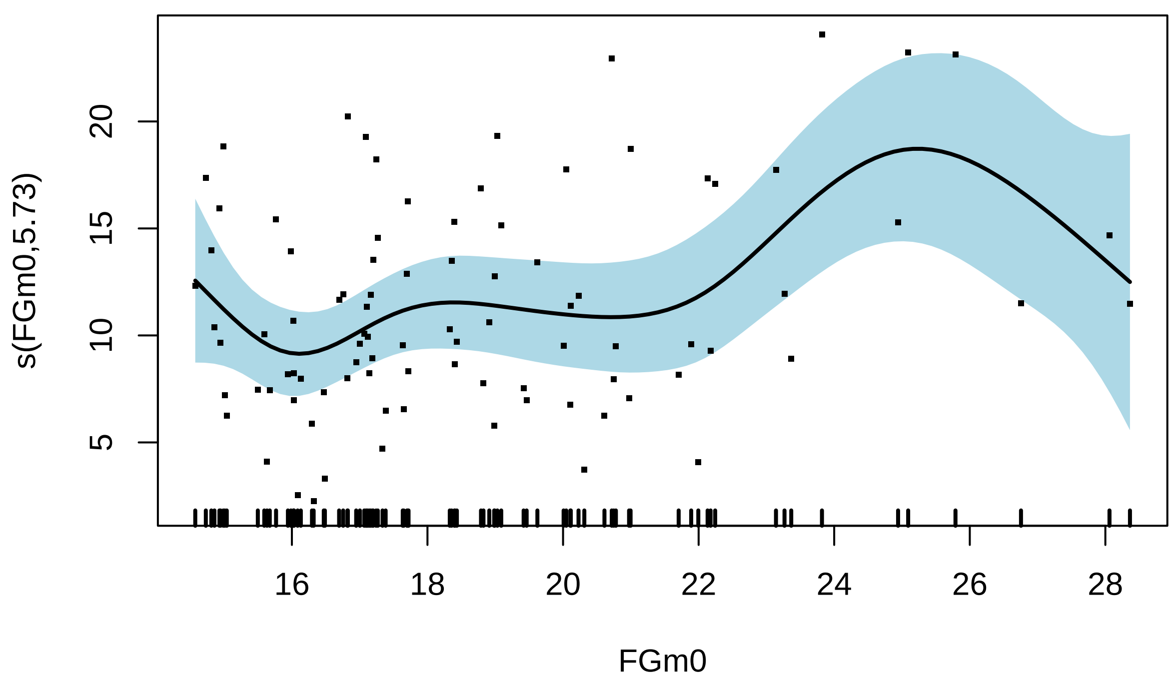
We also create a model like gam_1.1 but without Treatment as factor, to see what difference it makes.

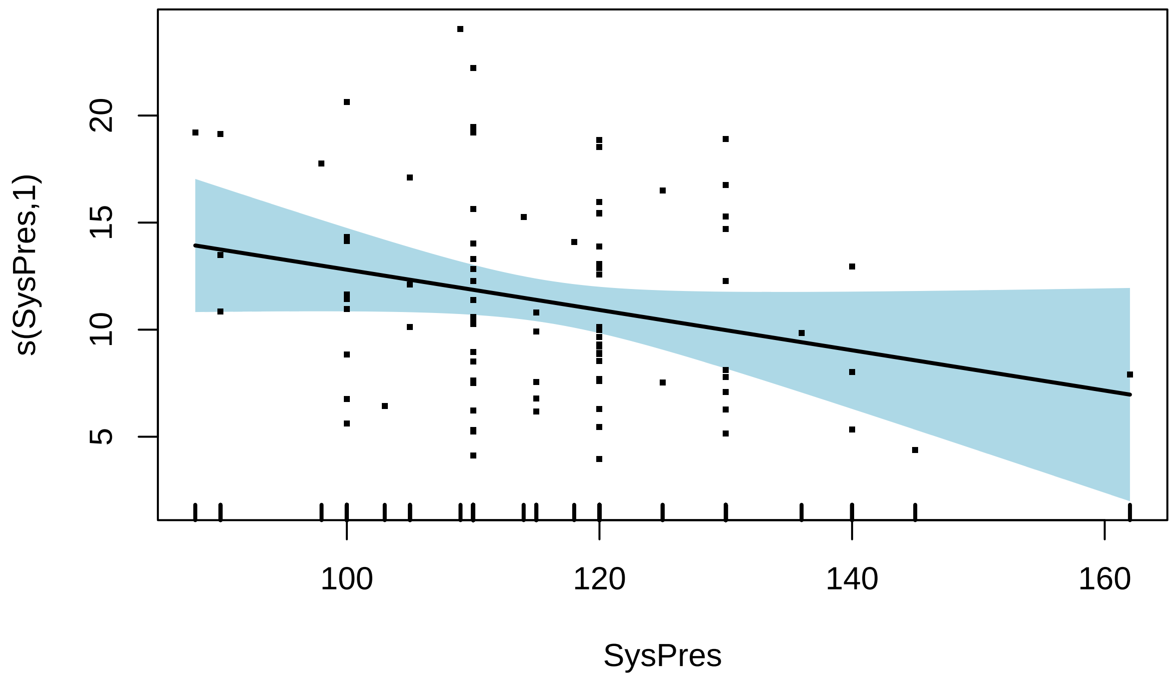
```
gam_1.2 <- gam(FGm12 ~ s(FGm0) + s(SysPres) + s(DiaPres) + s(height) + s(weight) )
summary(gam_1.2)
```

```
##
```

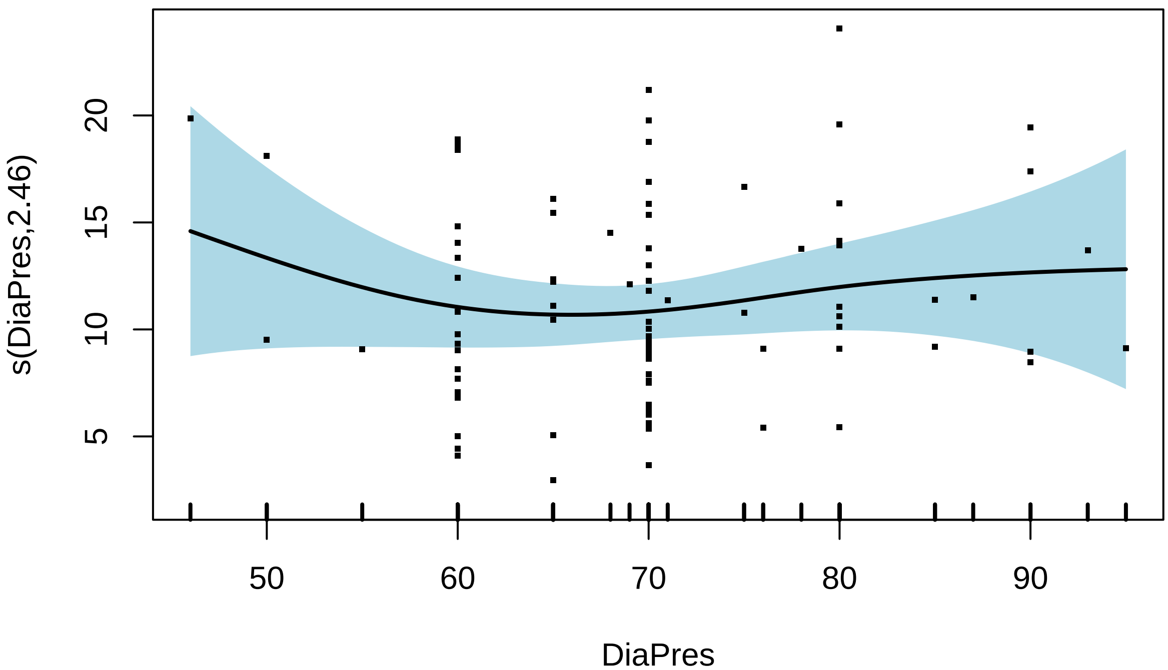
```
## Family: gaussian
## Link function: identity
##
## Formula:
## FGm12 ~ s(FGm0) + s(SysPres) + s(DiaPres) + s(height) + s(weight)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.0529     0.4949   18.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F p-value
## s(FGm0)    5.731  6.839 2.555  0.0187 *
## s(SysPres) 1.000  1.000 3.166  0.0791 .
## s(DiaPres) 2.456  3.082 0.983  0.4051
## s(height)  1.000  1.000 0.504  0.4800
## s(weight)  1.475  1.806 0.690  0.5879
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.184   Deviance explained =   29%
## GCV = 25.887  Scale est. = 22.285    n = 91
```

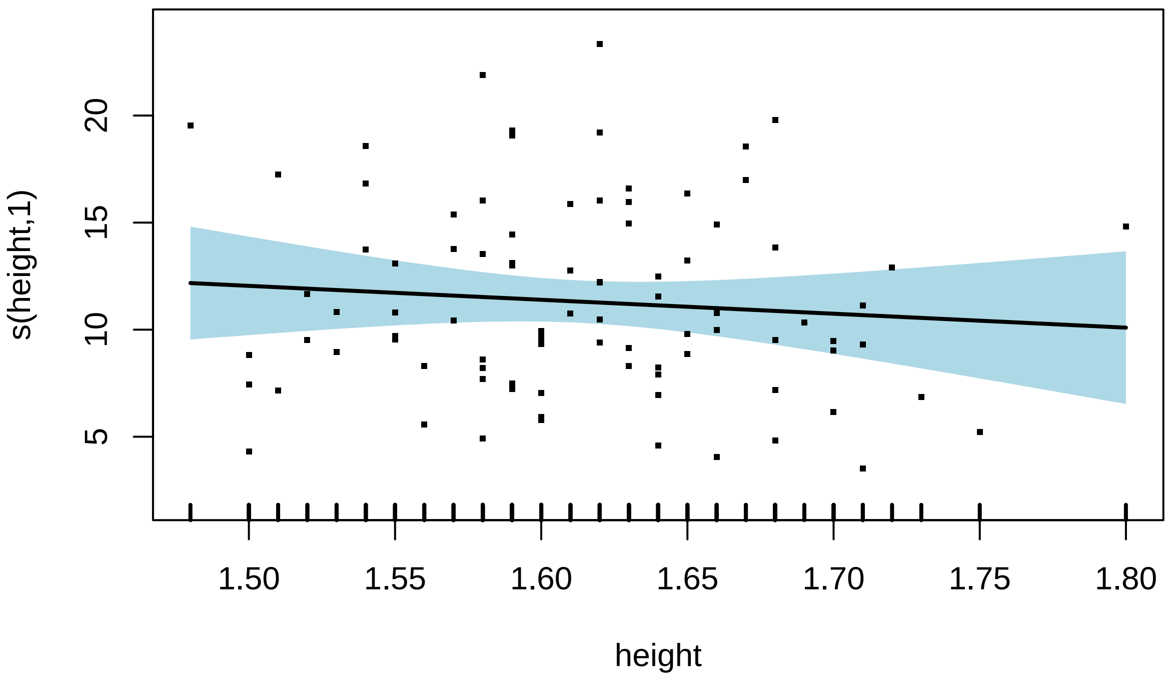
```r
plot(gam_1.2,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= co
```



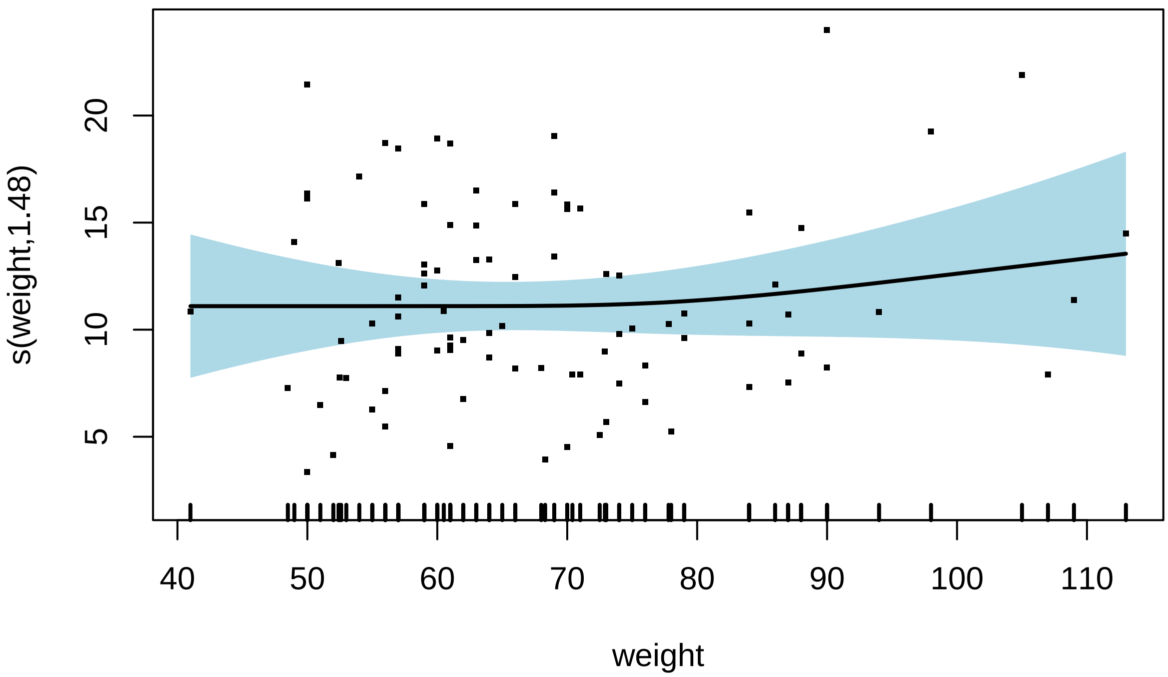


s(SysPres,1)
5
10
15
20
100
120
140
160
SysPres

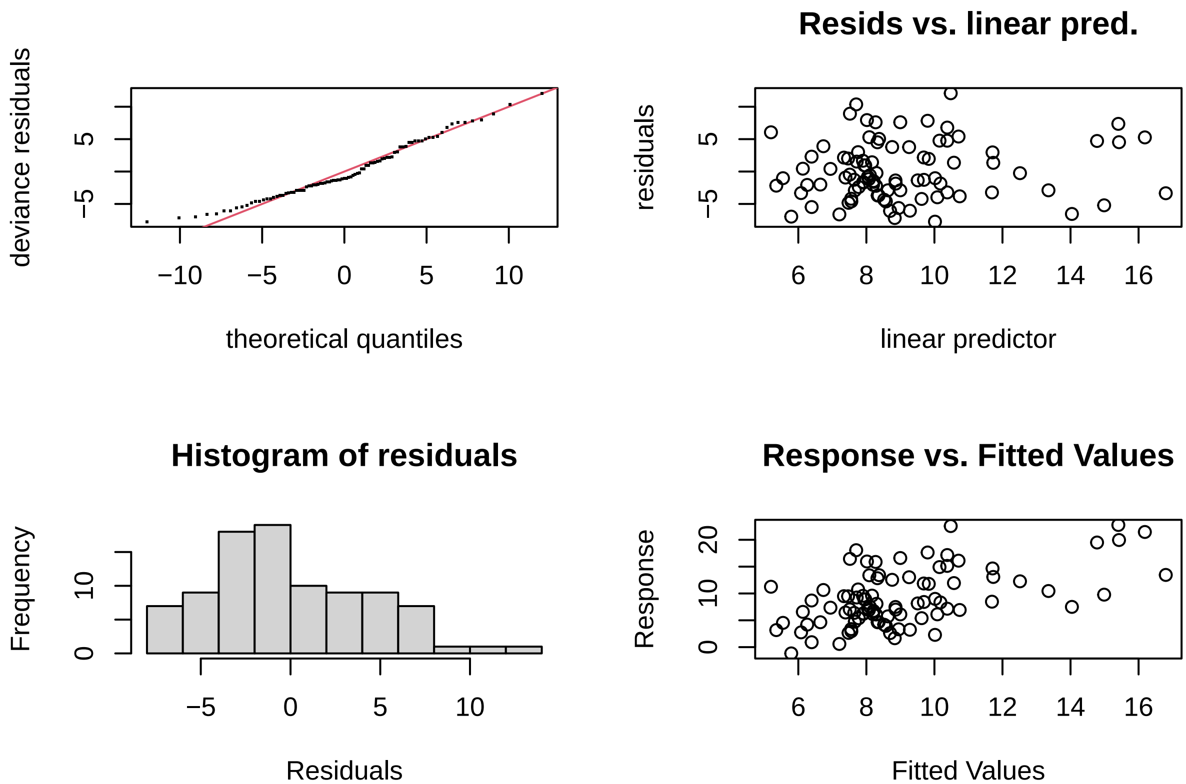

s(DiaPres,2.46)
5
10
15
20
50
60
70
80
90
DiaPres



s(height,1)
5
10
15
20
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height




```
gam.check(gam_1.2)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 47 iterations.
## The RMS GCV score gradient at convergence was 9.78451e-07 .
## The Hessian was positive definite.
## Model rank =  46 / 46
##
## 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(FGm0)    9.00 5.73    1.13    0.89
## s(SysPres) 9.00 1.00    0.98    0.44
## s(DiaPres) 9.00 2.46    0.90    0.18
## s(height)  9.00 1.00    0.90    0.18
## s(weight)  9.00 1.48    1.11    0.82
```

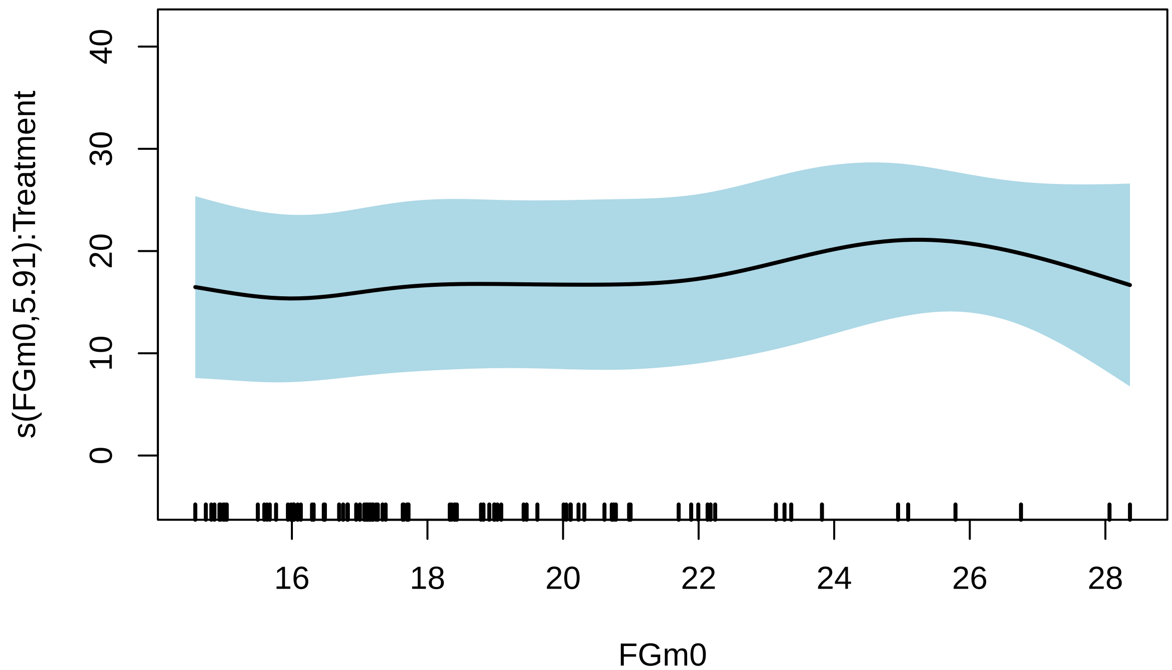
Now the partials show us less constant plots, but the deviance explained is less than using treatment as factor, letting us conclude that it is good too use Treatment as factor.

Further, we will create semiparametric models. We look at the effective degree of freedoms of model 1.1. SysPres, FGm0 and DiaPres have a edf close to 1 indicating a possible linear relationship to our prediction. It might improve the model to simplify it and incorporate this variables as linear terms. At first we set each of them linear respectively while letting the other parameters wrapped by a smoothing function.

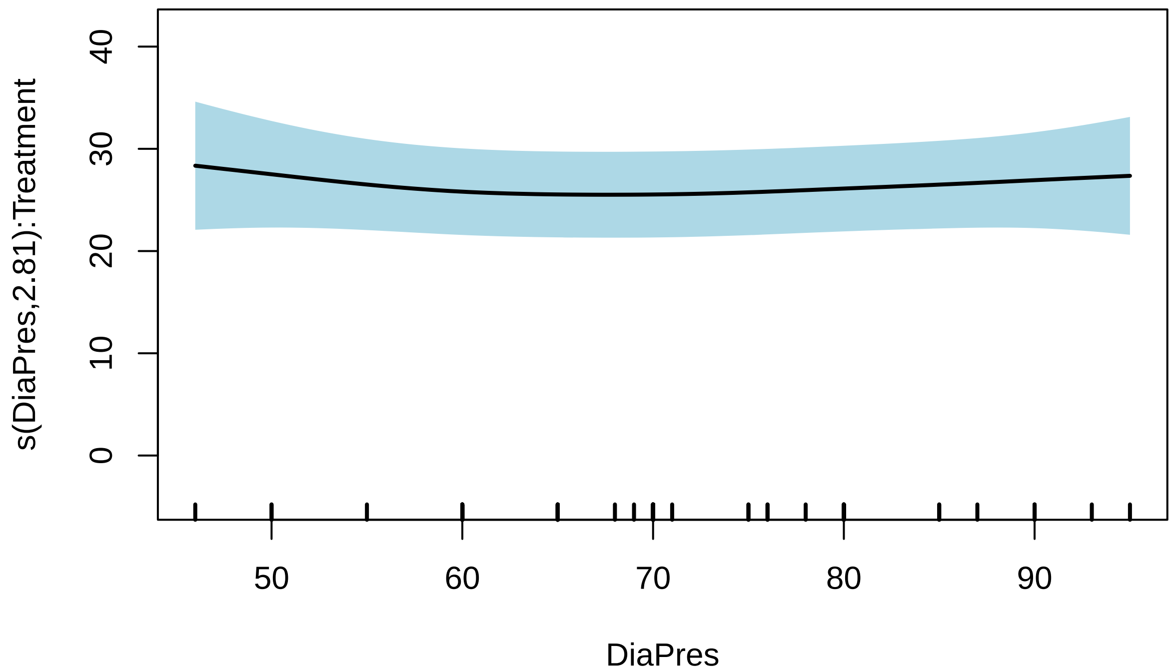
```r
# semiparametric model - SysPres (low df at model 1.1)
gam_2.1 <- gam(FGm12 ~ s(FGm0, by=Treatment) + SysPres + s(DiaPres, by=Treatment) + s(height, by=Treatm
summary(gam_2.1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## FGm12 ~ s(FGm0, by = Treatment) + SysPres + s(DiaPres, by = Treatment) +
##     s(height, by = Treatment) + s(weight, by = Treatment)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.02892    5.11037   4.506 2.29e-05 ***
## SysPres     -0.10944    0.04397  -2.489   0.0149 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                       edf Ref.df     F p-value
## s(FGm0):Treatment    5.905  6.996 3.748 0.00154 **
## s(DiaPres):Treatment 2.807  3.431 1.589 0.27970
## s(height):Treatment  1.250  1.250 2.233 0.31647
## s(weight):Treatment  1.250  1.250 0.417 0.43501
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 39/42
## R-sq.(adj) =  0.297   Deviance explained = 39.3%
## GCV = 22.438  Scale est. = 19.18     n = 91
```

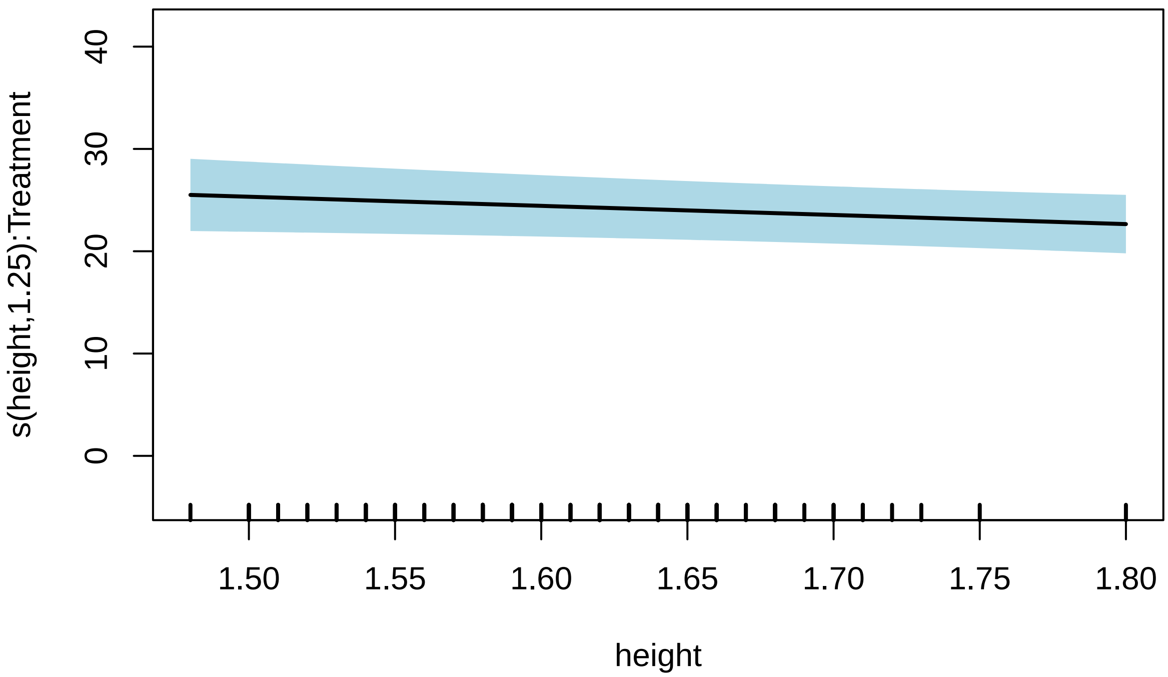
```r
plot(gam_2.1,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= c
```

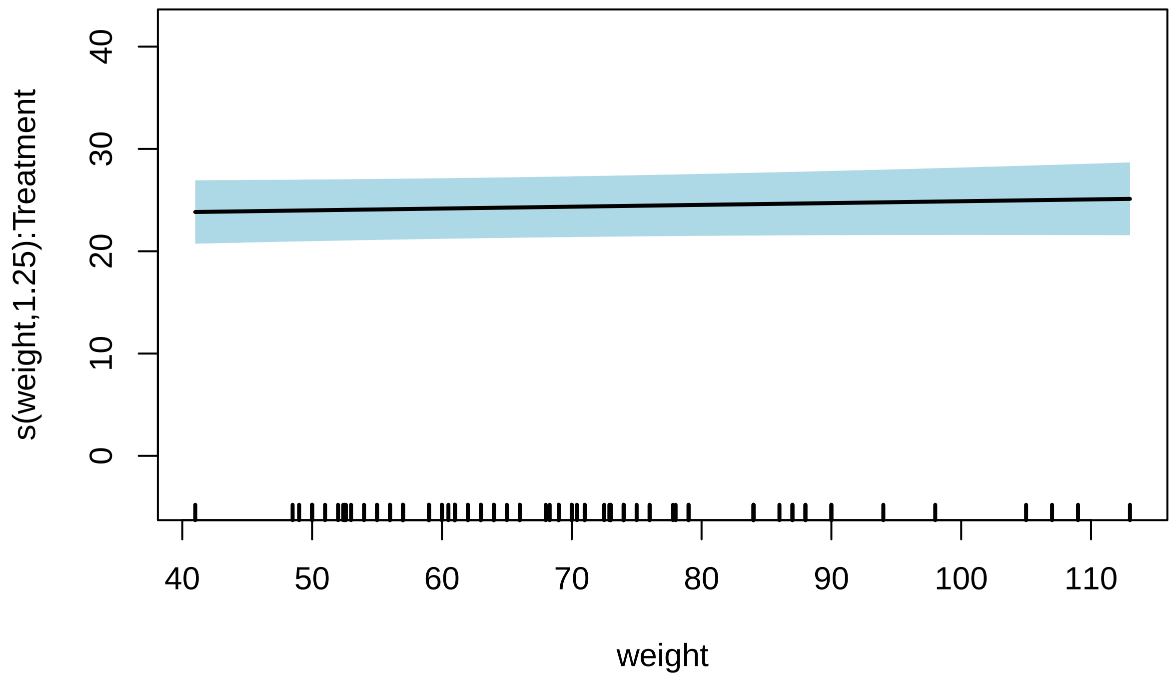


s(FGm0,5.91):Treatment
0
10
20
30
40
16
18
20
22
24
26
28
FGm0

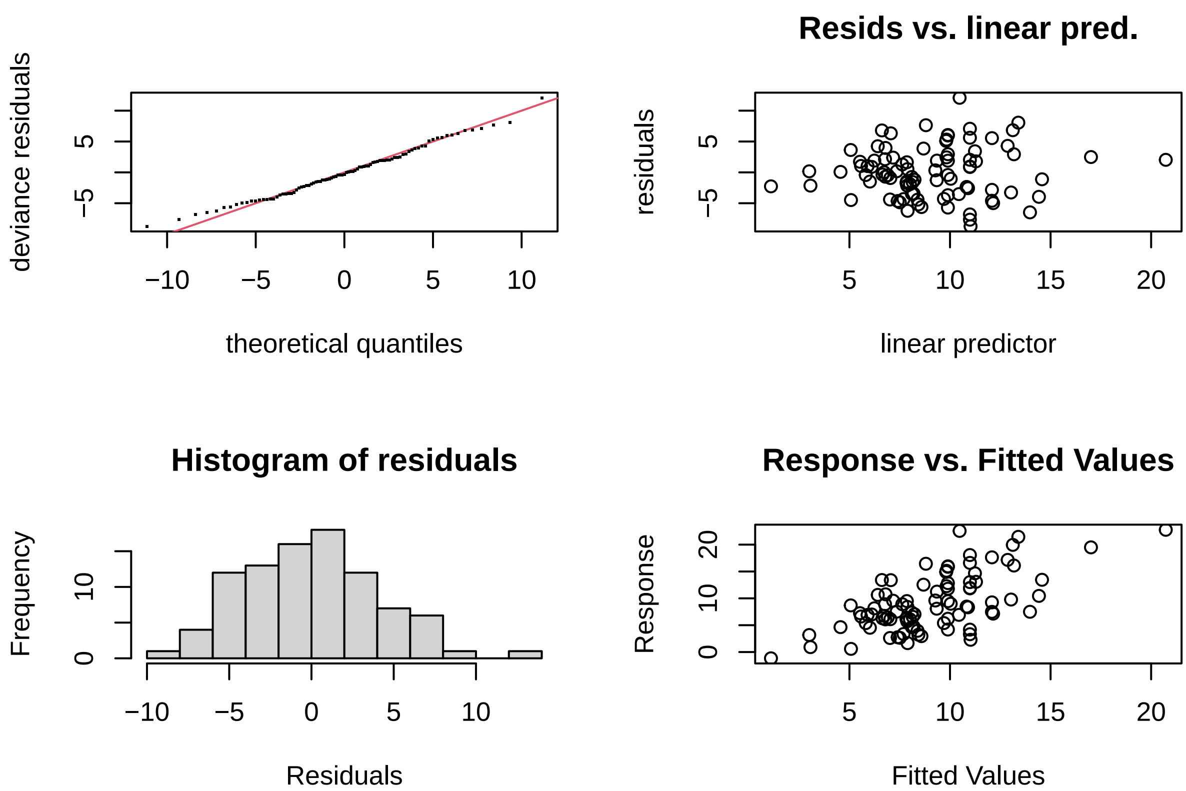


s(DiaPres,2.81):Treatment
0
10
20
30
40
50
60
70
80
90
DiaPres


s(height,1.25):Treatment
0
10
20
30
40
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height




```r
gam.check(gam_2.1)
```



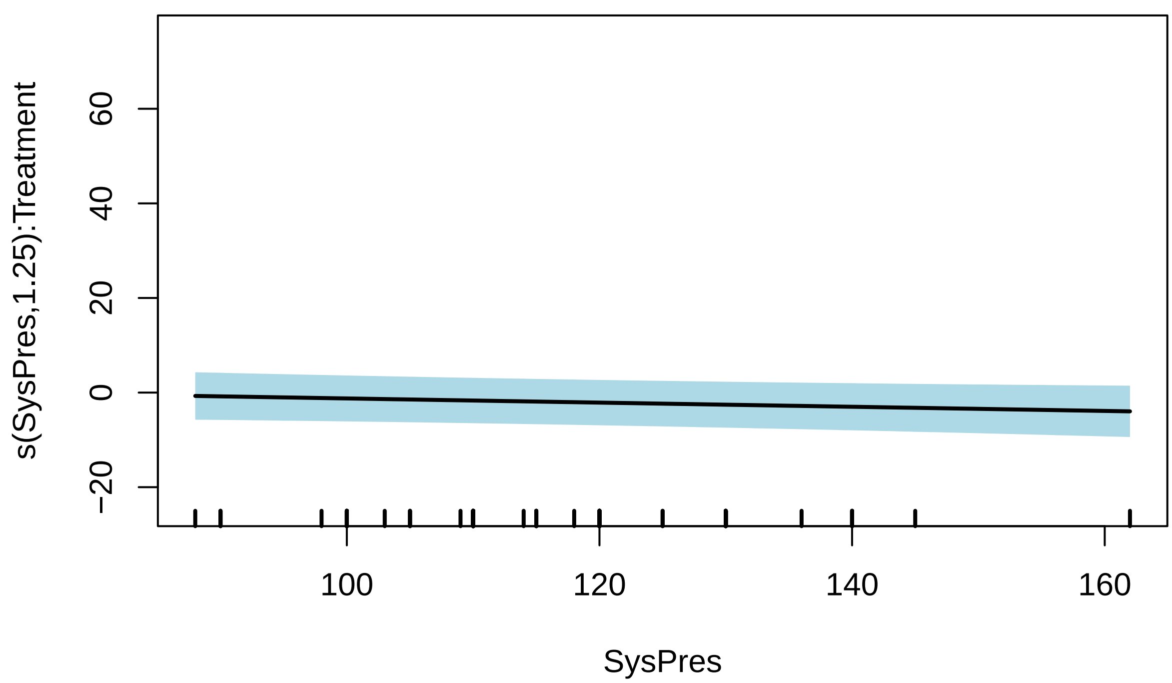

```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 17 iterations.
## The RMS GCV score gradient at convergence was 9.613396e-06 .
## The Hessian was positive definite.
## Model rank =  39 / 42
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                            k'   edf k-index p-value
## s(FGm0):Treatment    10.00  5.91    1.13    0.84
## s(DiaPres):Treatment 10.00  2.81    0.96    0.32
## s(height):Treatment  10.00  1.25    0.85    0.08 .
## s(weight):Treatment  10.00  1.25    1.15    0.87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

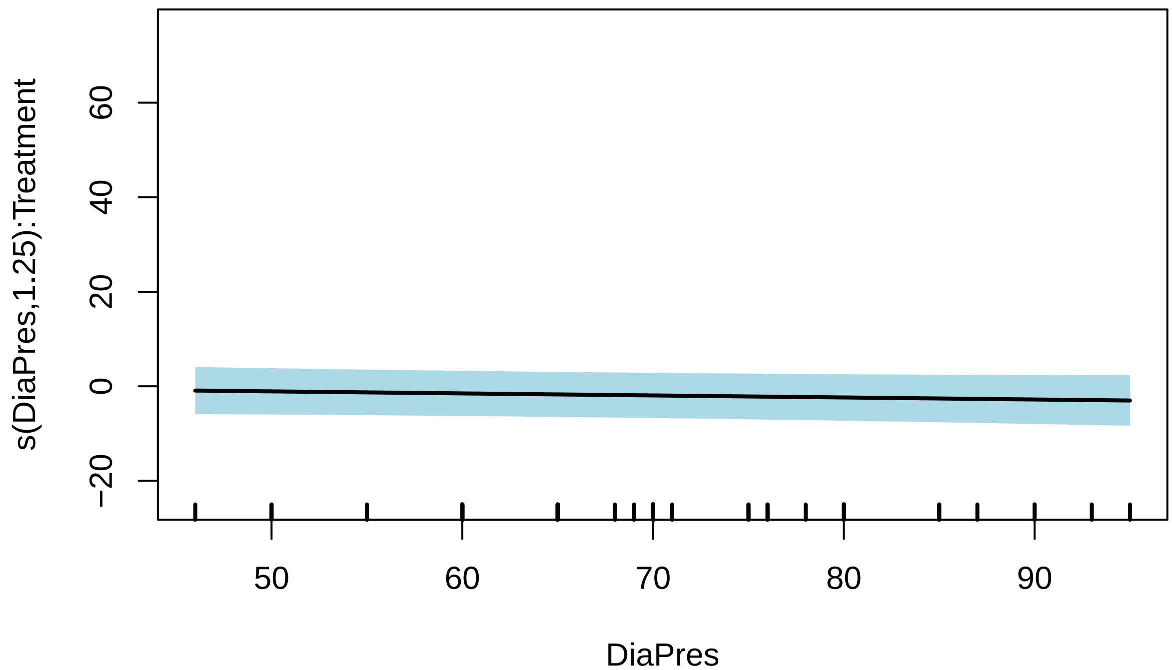
```
# semiparametric model - FGm0 (low df at model 1.1)
gam_2.2 <- gam(FGm12 ~ FGm0 + s(SysPres, by=Treatment) + s(DiaPres, by=Treatment) + s(height, by=Treatme
summary(gam_2.2)
```

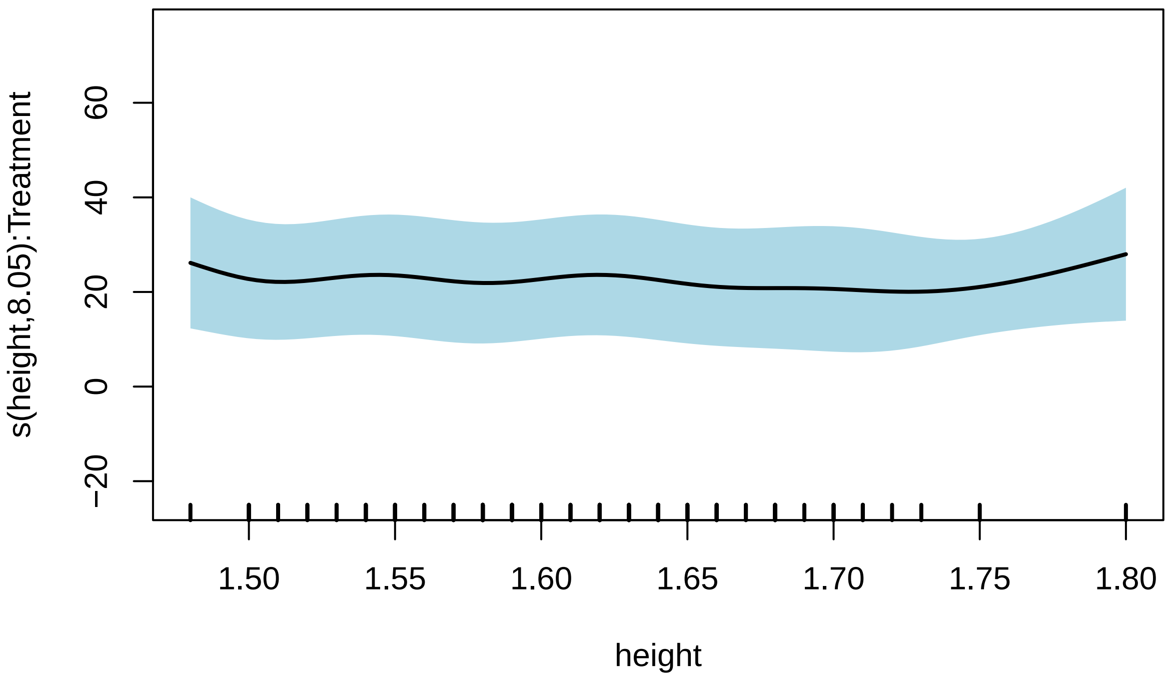
```
##
## Family: gaussian
## Link function: identity
```

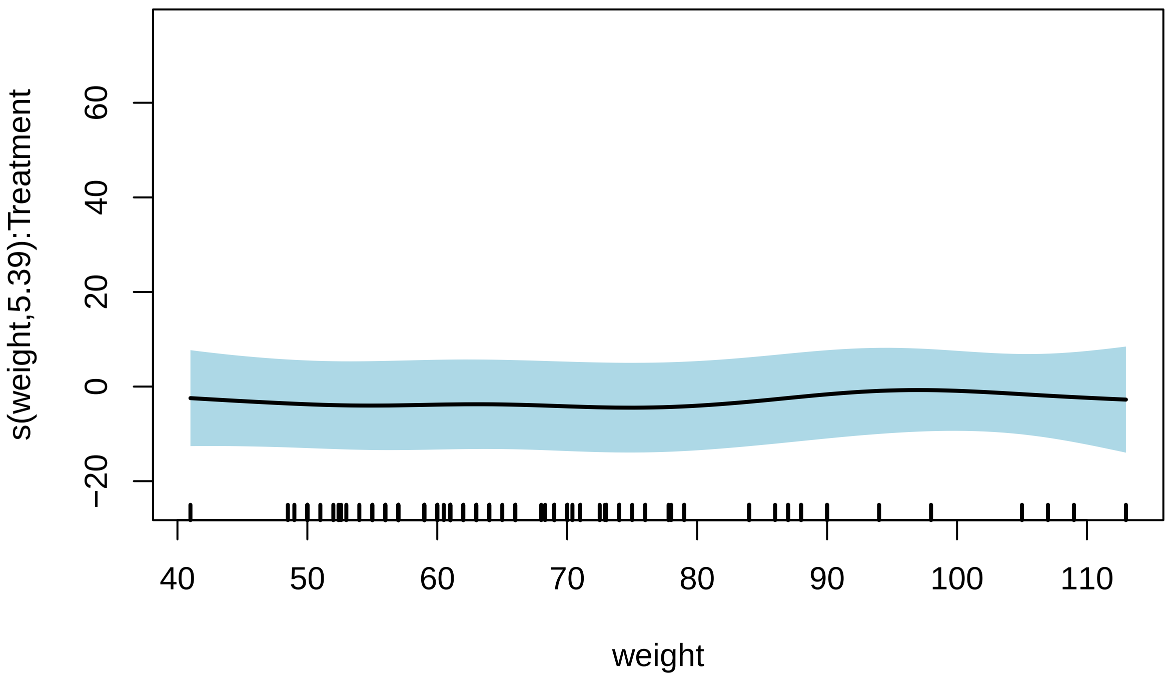
```
##
## Formula:
## FGm12 ~ FGm0 + s(SysPres, by = Treatment) + s(DiaPres, by = Treatment) +
##     s(height, by = Treatment) + s(weight, by = Treatment)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.1985     3.1673   1.326   0.1891
## FGm0          0.3935     0.1678   2.345   0.0217 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df     F p-value
## s(SysPres):Treatment 1.250  1.250 3.752 0.02580 *
## s(DiaPres):Treatment 1.250  1.250 6.488 0.02072 *
## s(height):Treatment  8.053  8.807 3.325 0.00312 **
## s(weight):Treatment  5.385  6.463 1.992 0.07856 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 39/42
## R-sq.(adj) =  0.352   Deviance explained = 47.4%
## GCV = 22.034  Scale est. = 17.691    n = 91
```

```r
plot(gam_2.2,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= c
```

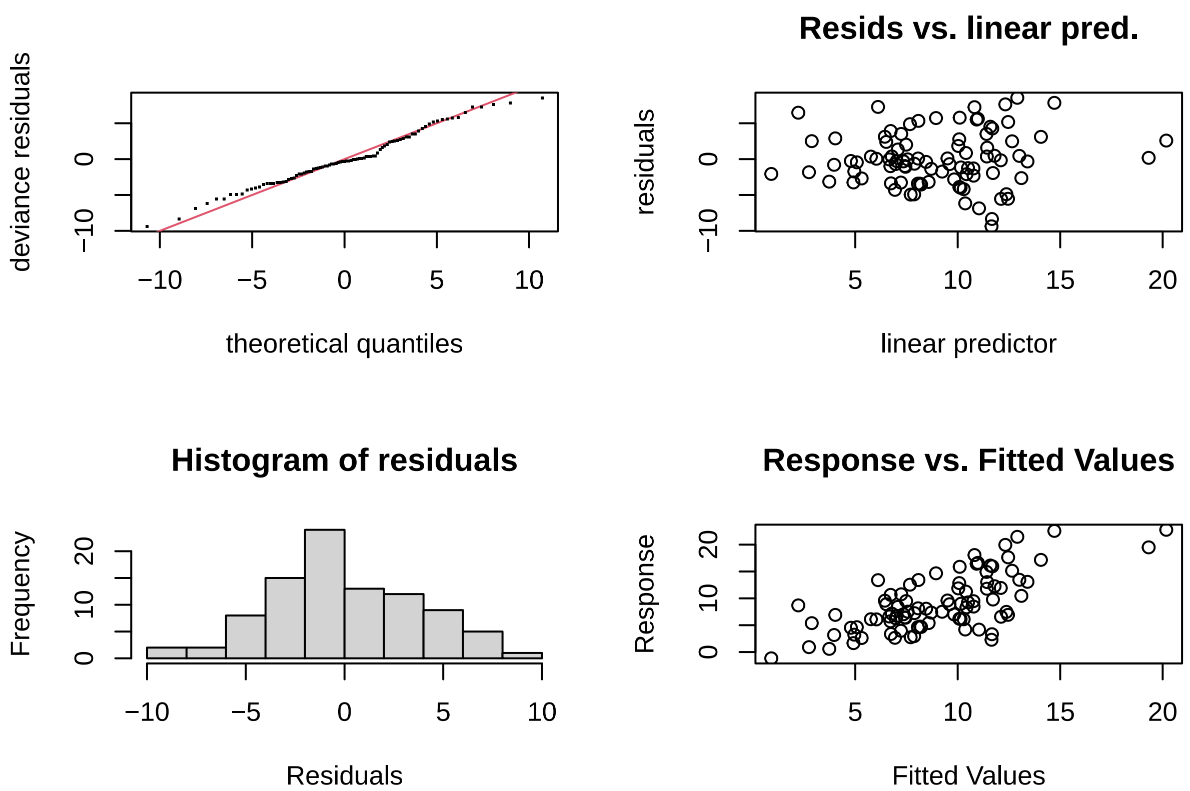



s(DiaPres,1.25):Treatment
−20
0
20
40
60
50
60
70
80
90
DiaPres


s(height,8.05):Treatment
−20
0
20
40
60
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height




```r
gam.check(gam_2.2)
```



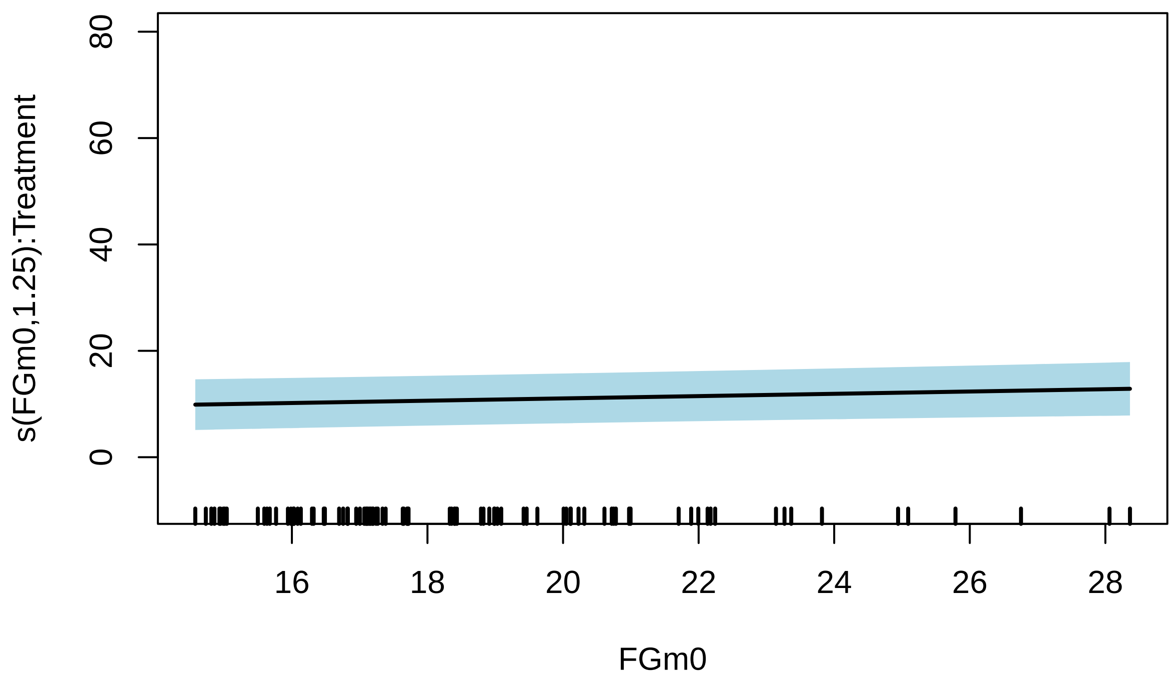

```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 19 iterations.
## The RMS GCV score gradient at convergence was 5.921529e-07 .
## The Hessian was positive definite.
## Model rank =  39 / 42
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                          k'   edf k-index p-value
## s(SysPres):Treatment 10.00  1.25    1.07    0.68
## s(DiaPres):Treatment 10.00  1.25    1.05    0.67
## s(height):Treatment  10.00  8.05    0.92    0.20
## s(weight):Treatment  10.00  5.39    1.06    0.65
```

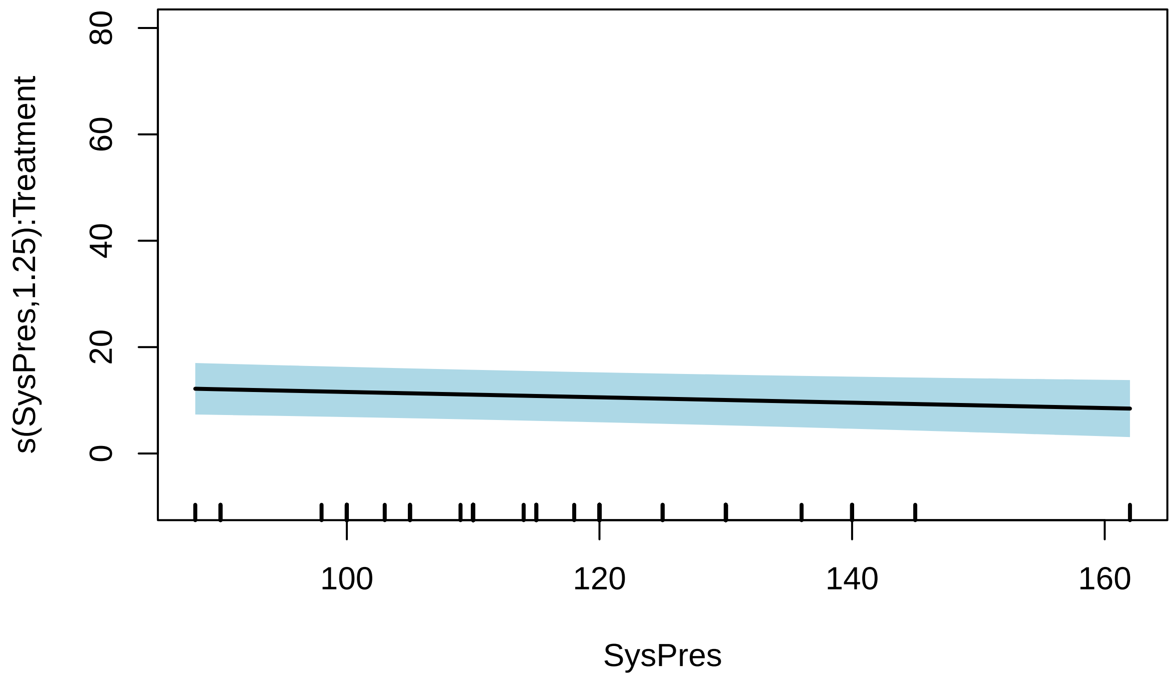
```r
# semiparametric model - DiaPres (low df at model 1.1)
gam_2.3 <- gam(FGm12 ~ s(FGm0, by=Treatment) + s(SysPres, by=Treatment) + DiaPres + s(height, by=Treatm
summary(gam_2.3)
```

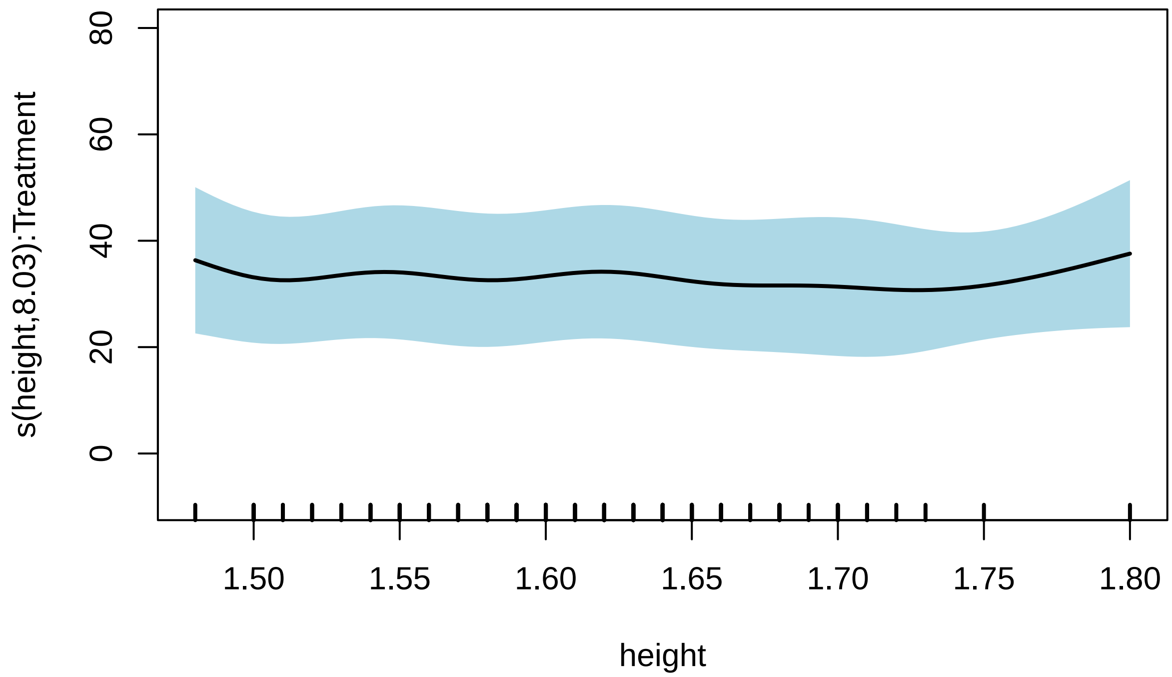
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
```

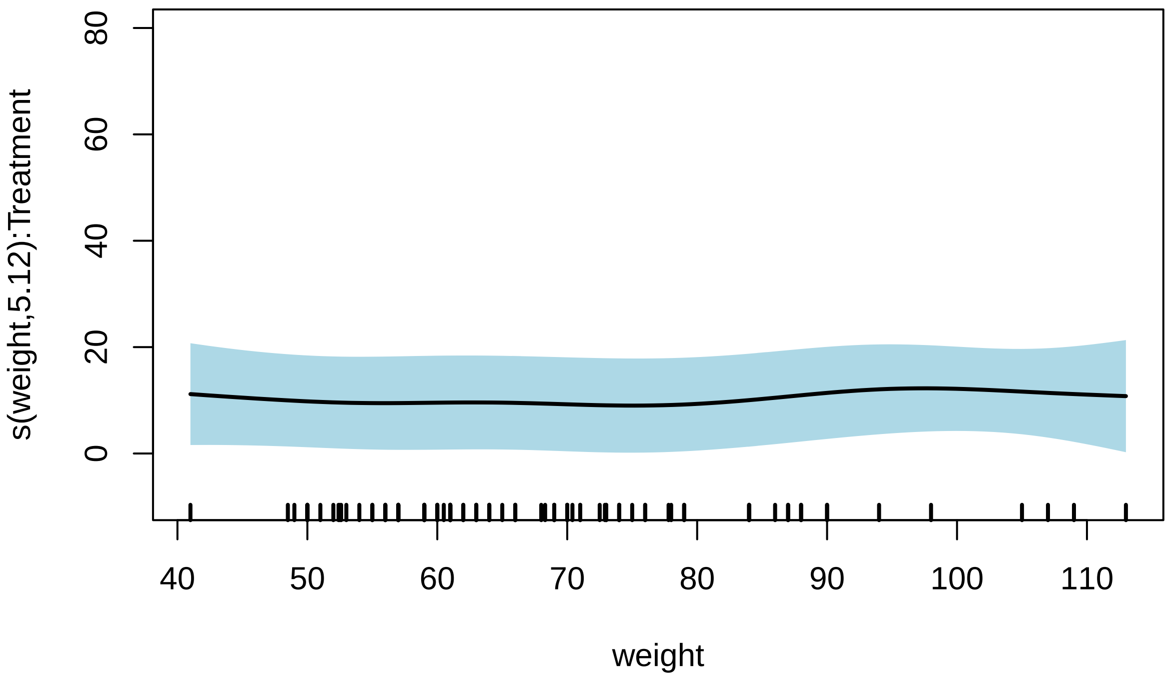
```
## FGm12 ~ s(FGm0, by = Treatment) + s(SysPres, by = Treatment) +
##     DiaPres + s(height, by = Treatment) + s(weight, by = Treatment)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.48590    4.26759   3.863 0.000239 ***
## DiaPres     -0.07668    0.06110  -1.255 0.213491
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df     F p-value
## s(FGm0):Treatment     1.250  1.250 8.305  0.0371 *
## s(SysPres):Treatment 1.250  1.250 3.254  0.0348 *
## s(height):Treatment  8.029  8.789 2.546  0.0182 *
## s(weight):Treatment  5.124  6.174 1.859  0.0976 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 39/42
## R-sq.(adj) =  0.347   Deviance explained = 46.8%
## GCV = 22.109  Scale est. = 17.82     n = 91
```

```r
plot(gam_2.3,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= co
```

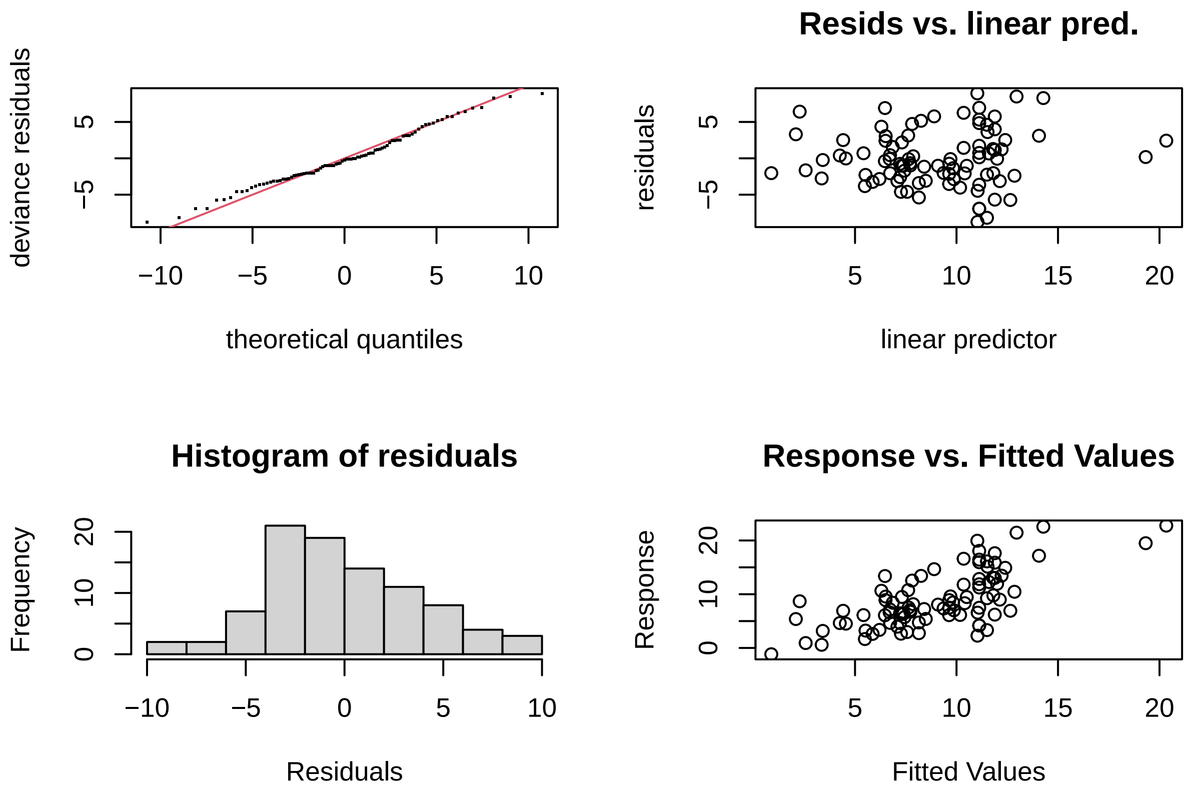



s(SysPres,1.25):Treatment
0
20
40
60
80
100
120
140
160
SysPres


s(height,8.03):Treatment
0
20
40
60
80
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height




```
gam.check(gam_2.3)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 19 iterations.
## The RMS GCV score gradient at convergence was 8.200385e-07 .
## The Hessian was positive definite.
## Model rank =  39 / 42
##
## 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(FGm0):Treatment     10.00  1.25    1.10    0.78
## s(SysPres):Treatment 10.00  1.25    1.05    0.60
## s(height):Treatment  10.00  8.03    0.93    0.20
## s(weight):Treatment  10.00  5.12    1.06    0.64
```

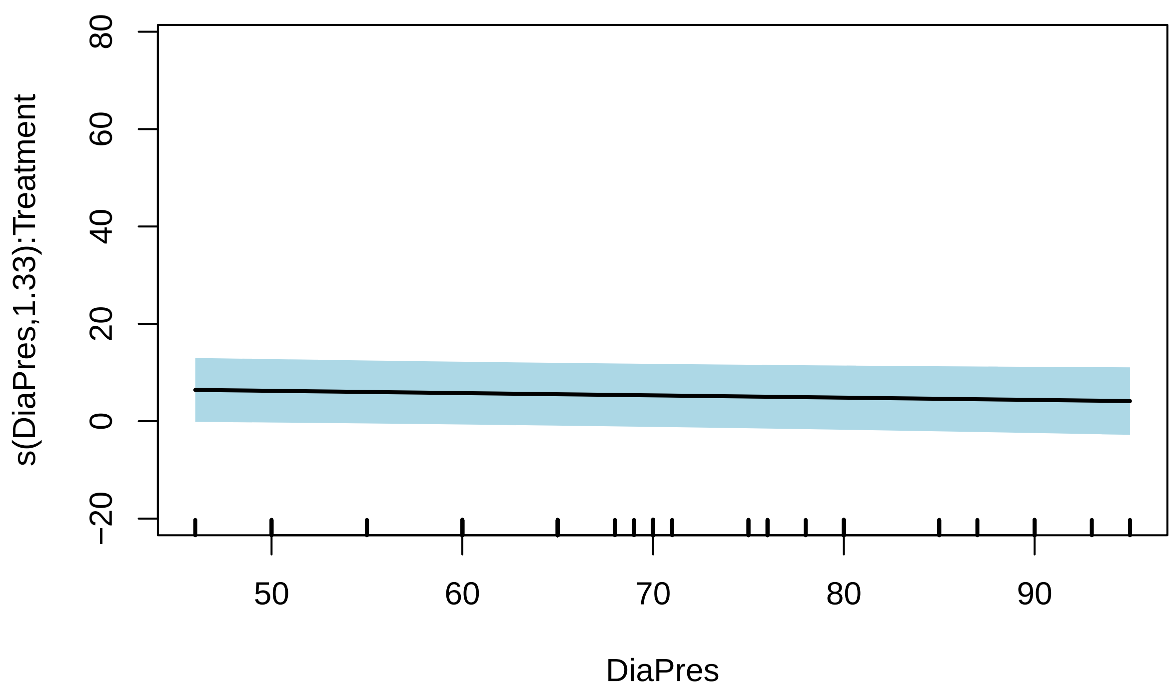
We also set two of them linear:

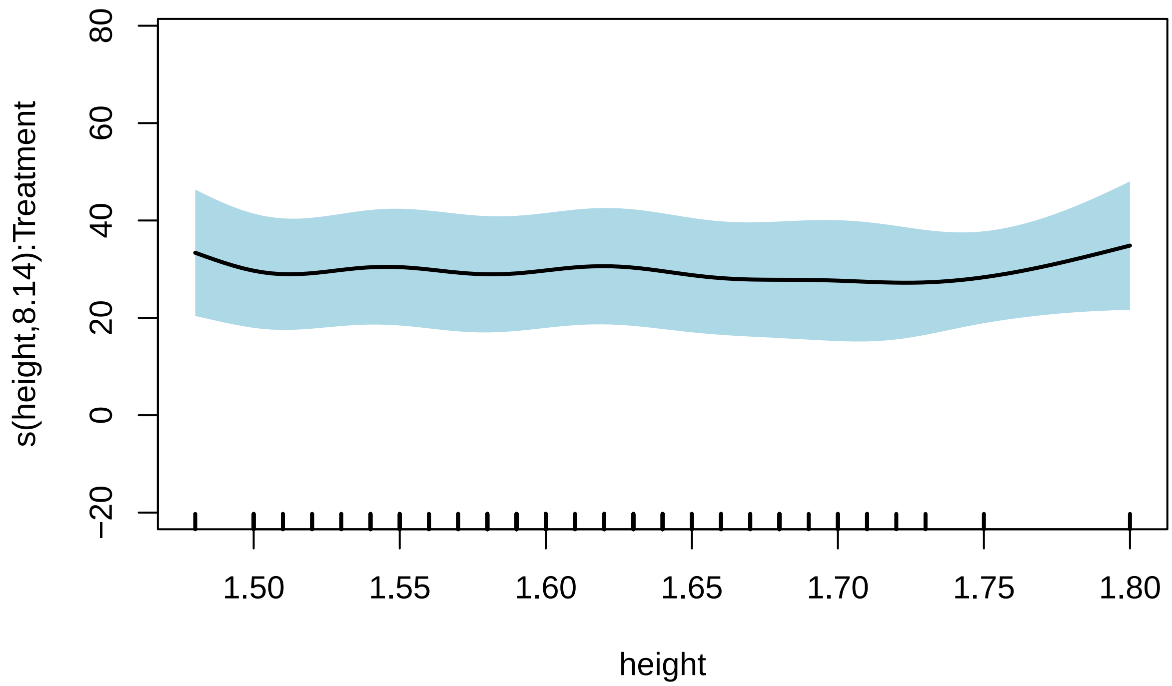
```
# semiparametric model - FGm0 + SysPres
gam_3.1 <- gam(FGm12 ~ FGm0 + SysPres + s(DiaPres, by=Treatment) + s(height, by=Treatment) + s(weight, 
summary(gam_3.1)
```

```
##
## Family: gaussian
## Link function: identity
```

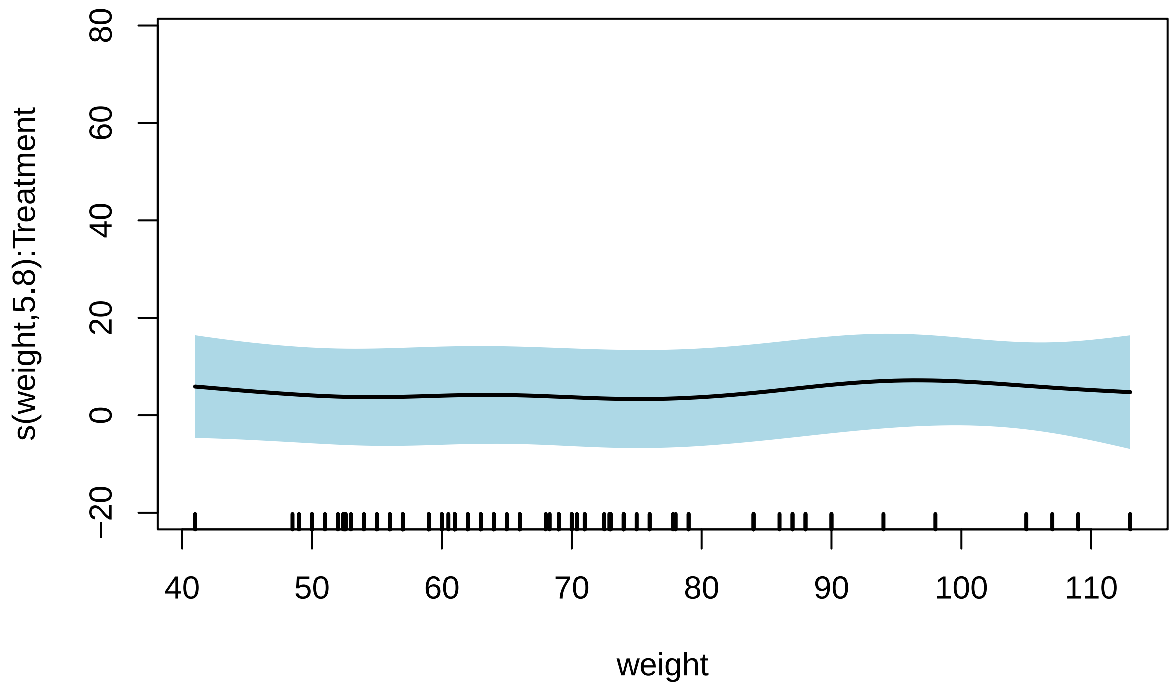
```
##
## Formula:
## FGm12 ~ FGm0 + SysPres + s(DiaPres, by = Treatment) + s(height,
##     by = Treatment) + s(weight, by = Treatment)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.56312    6.50579   2.085   0.0406 *
## FGm0         0.33678    0.17091   1.971   0.0526 .
## SysPres     -0.07329    0.04411  -1.662   0.1009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df     F p-value
## s(DiaPres):Treatment 1.333  1.333 6.359 0.02466 *
## s(height):Treatment  8.141  8.893 3.220 0.00393 **
## s(weight):Treatment  5.796  6.895 2.087 0.06228 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 31/33
## R-sq.(adj) =  0.359   Deviance explained = 48.2%
## GCV = 21.882  Scale est. = 17.489    n = 91
```

```
plot(gam_3.1,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= co
```

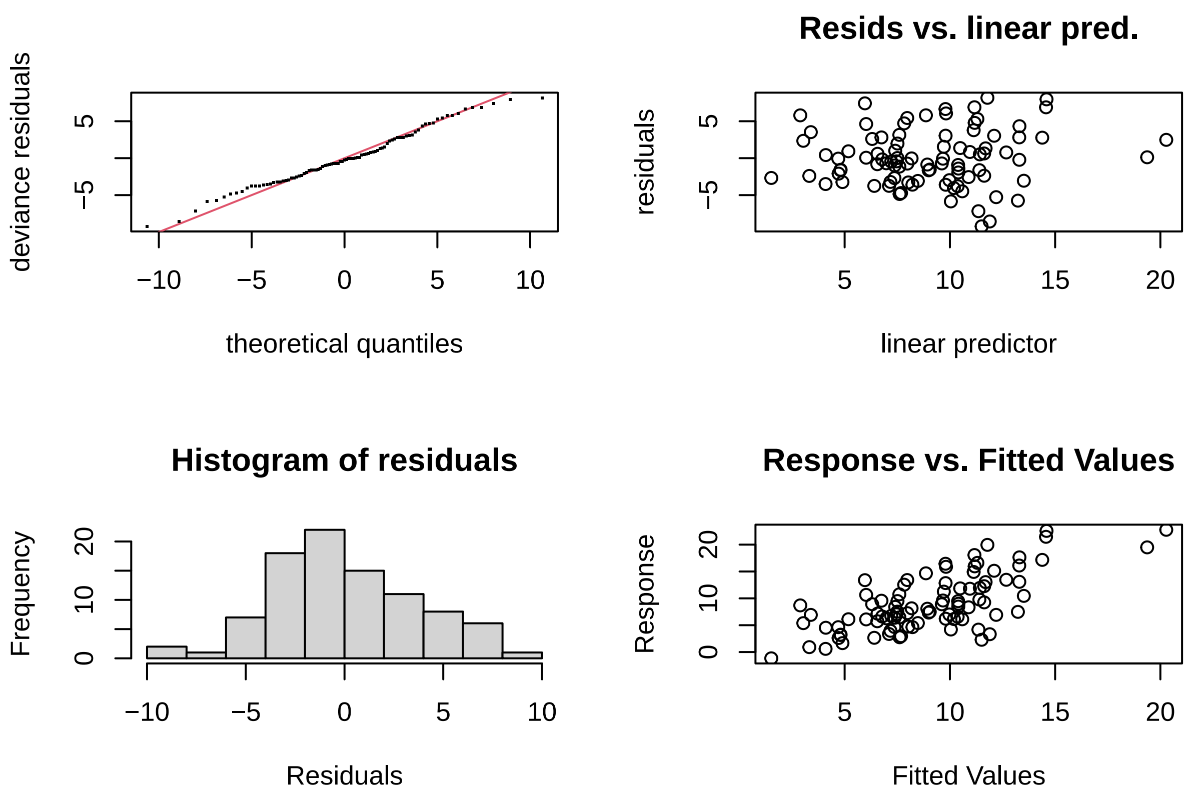




s(height,8.14):Treatment
80
60
40
20
0
−20
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height




```r
gam.check(gam_3.1)
```



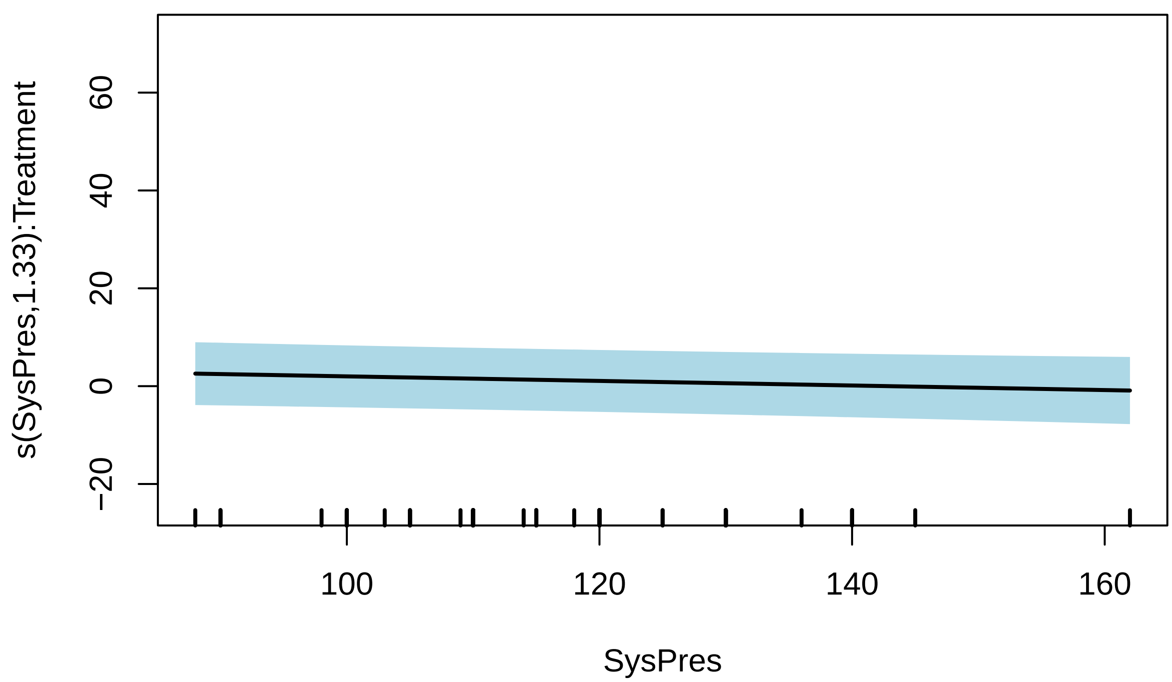

```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 18 iterations.
## The RMS GCV score gradient at convergence was 1.115215e-06 .
## The Hessian was positive definite.
## Model rank =  31 / 33
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                             k'   edf k-index p-value
## s(DiaPres):Treatment 10.00  1.33    1.01    0.48
## s(height):Treatment  10.00  8.14    0.91    0.18
## s(weight):Treatment  10.00  5.80    1.07    0.72
```

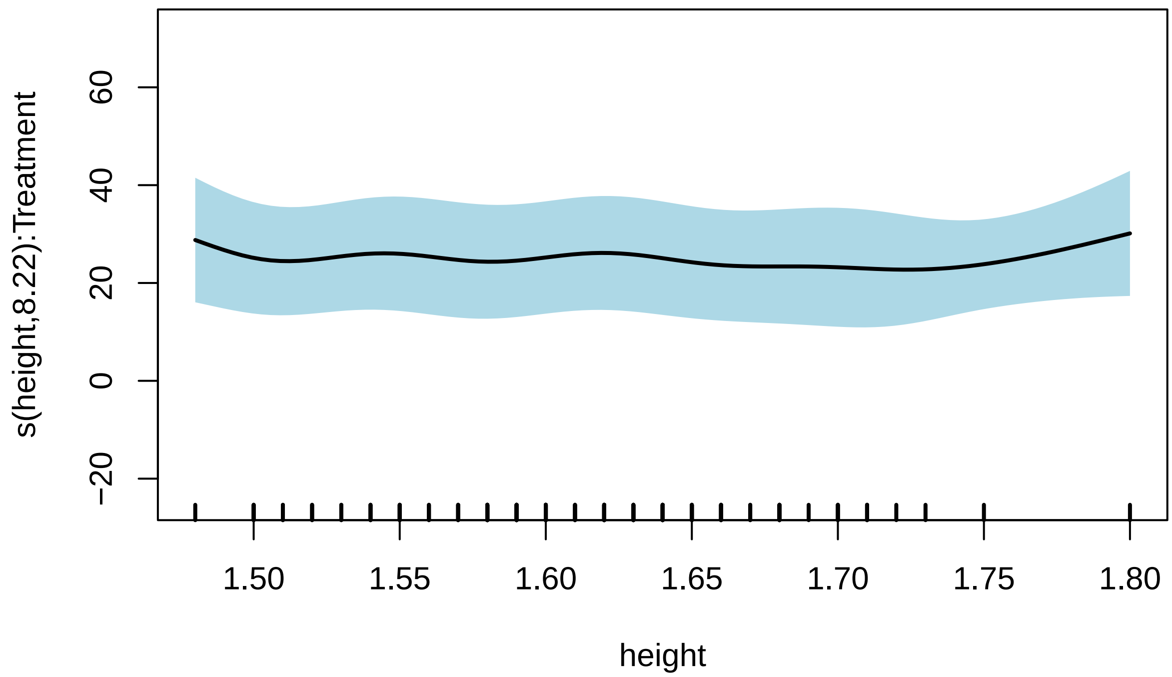
```
# semiparametric model - FGm0 + DiaPres
gam_3.2 <- gam(FGm12 ~ FGm0 + s(SysPres, by=Treatment) + DiaPres + s(height, by=Treatment) + s(weight, 
summary(gam_3.2)
```

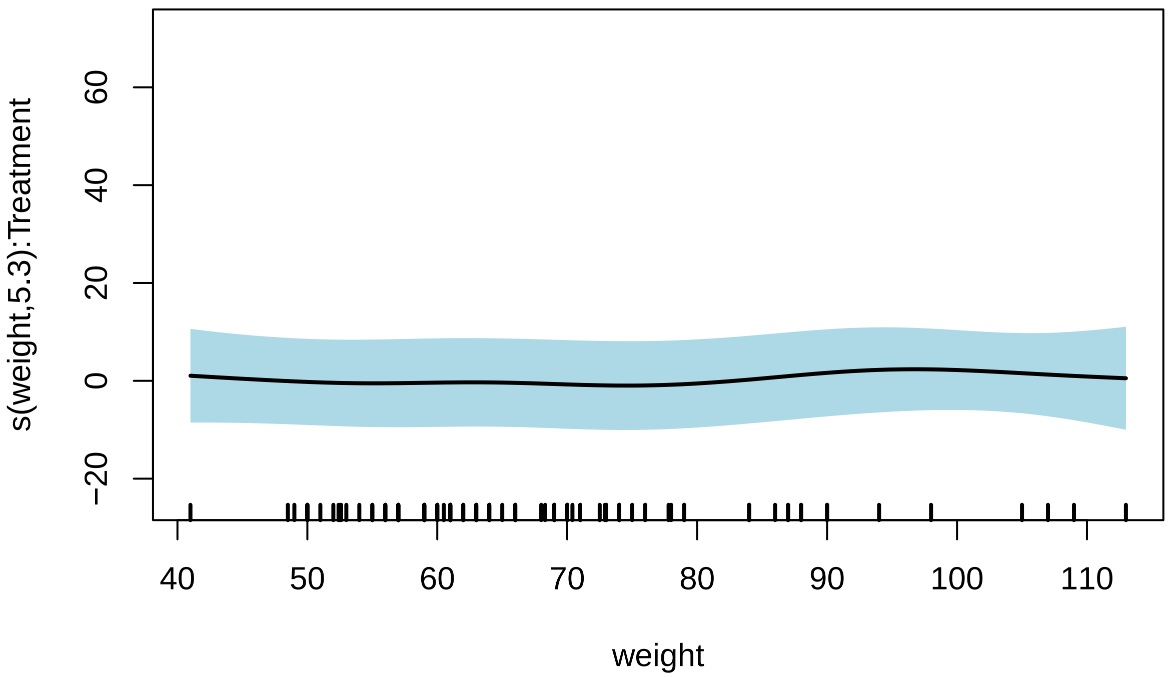
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## FGm12 ~ FGm0 + s(SysPres, by = Treatment) + DiaPres + s(height,
```

```
##     by = Treatment) + s(weight, by = Treatment)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.25036    5.59096   1.655   0.1023
## FGm0         0.36385    0.16917   2.151   0.0348 *
## DiaPres     -0.06703    0.06142  -1.091   0.2787
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df     F p-value
## s(SysPres):Treatment 1.333  1.333 3.374 0.03057 *
## s(height):Treatment  8.222  8.940 3.242 0.00309 **
## s(weight):Treatment  5.295  6.353 2.127 0.06460 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 31/33
## R-sq.(adj) =   0.35   Deviance explained = 47.2%
## GCV = 22.063  Scale est. = 17.735    n = 91
```

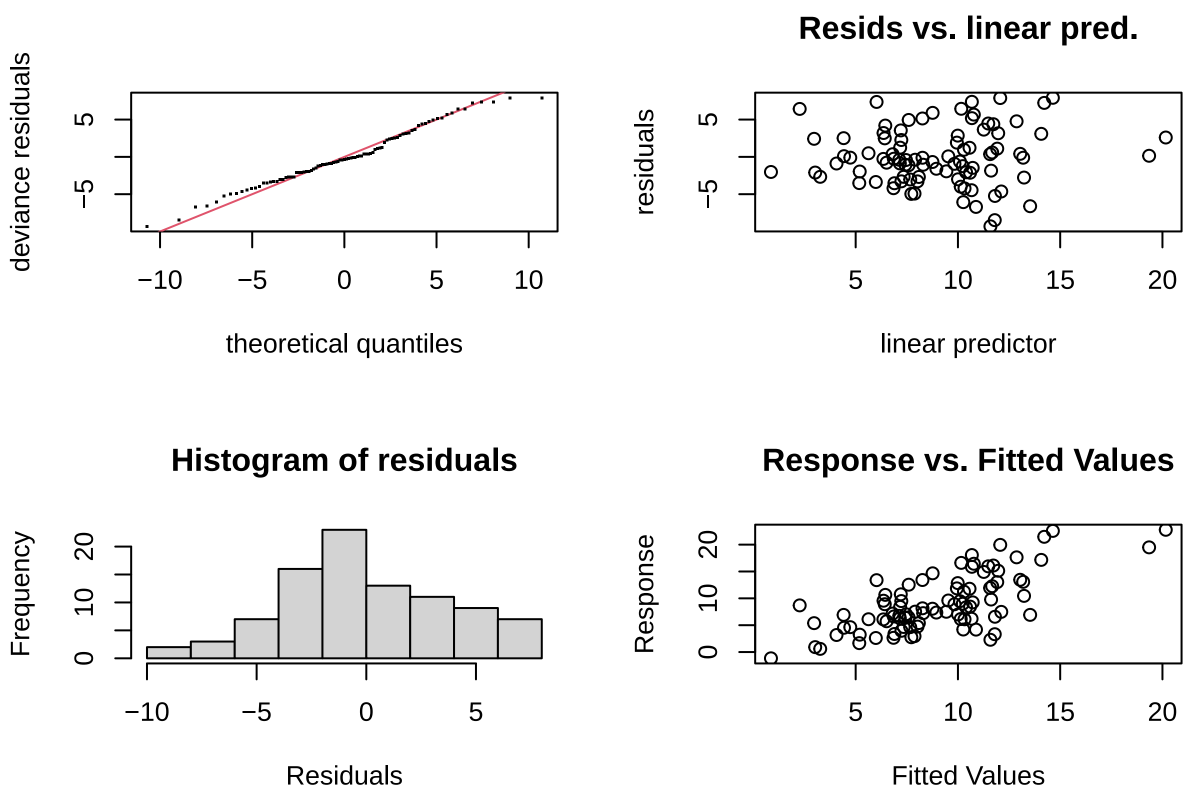
```r
plot(gam_3.2,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= c
```




s(height,8.22):Treatment
-20
0
20
40
60
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height




```r
gam.check(gam_3.2)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 17 iterations.
## The RMS GCV score gradient at convergence was 9.737704e-07 .
## The Hessian was positive definite.
## Model rank =  31 / 33
##
## 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(SysPres):Treatment 10.00  1.33    1.08    0.67
## s(height):Treatment  10.00  8.22    0.91    0.15
## s(weight):Treatment  10.00  5.30    1.06    0.69
```

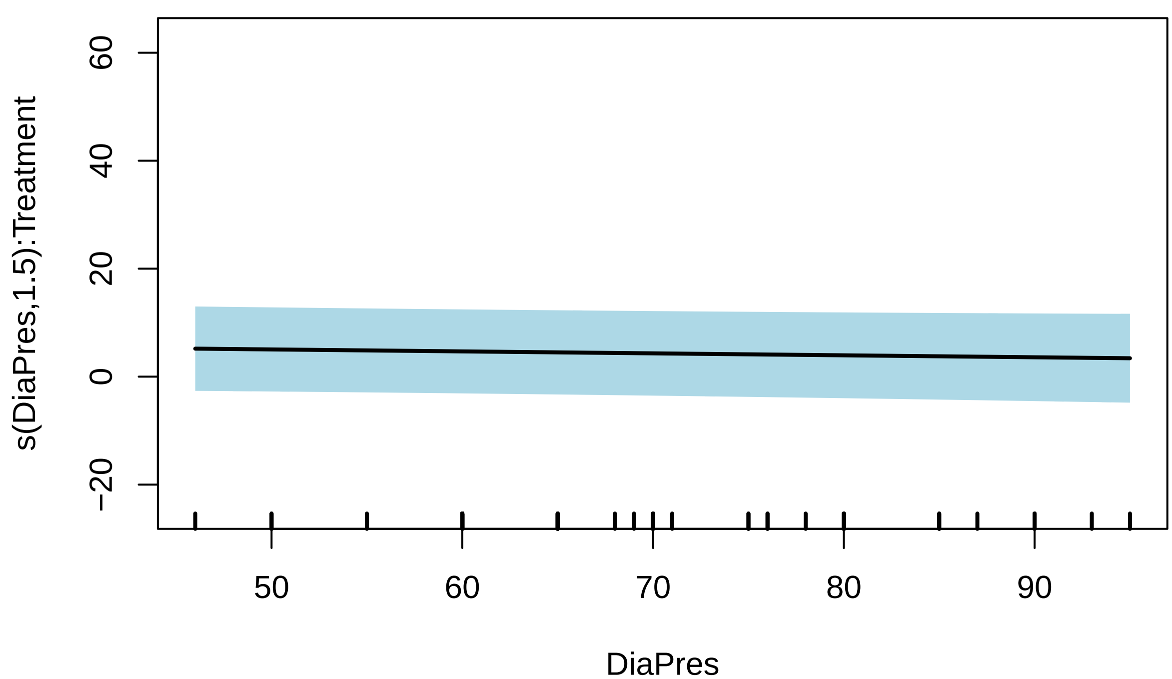
Here we removed the factor from weight.

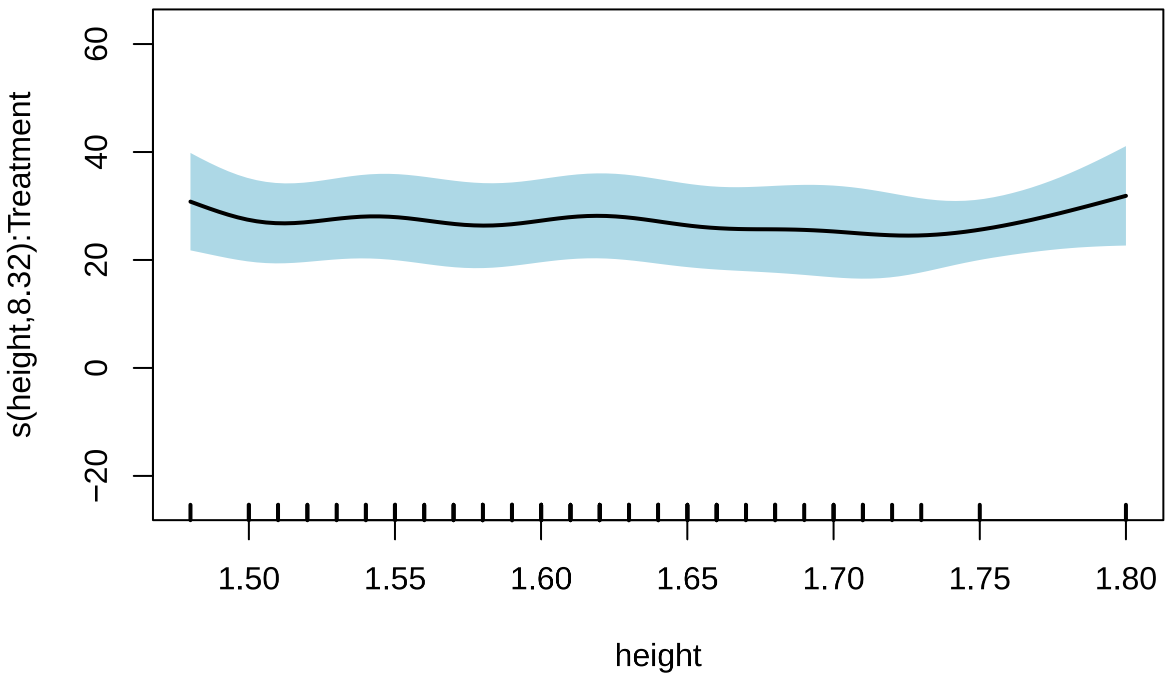
```
gam_4.1 <- gam(FGm12 ~ FGm0 + SysPres + s(DiaPres, by=Treatment) + s(height, by=Treatment) + s(weight))
summary(gam_4.1)
```

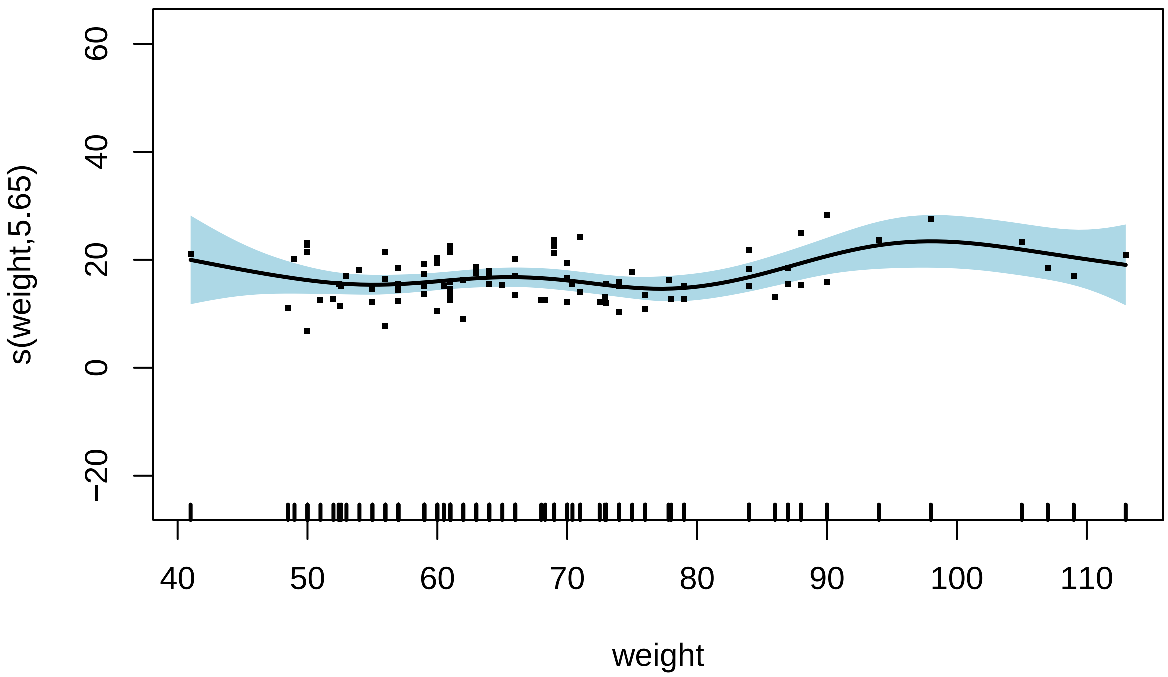
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
```

```
## FGm12 ~ FGm0 + SysPres + s(DiaPres, by = Treatment) + s(height,
##     by = Treatment) + s(weight)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.57911    6.79934   2.438   0.0172 *
## FGm0         0.32584    0.16961   1.921   0.0586 .
## SysPres     -0.09566    0.04566  -2.095   0.0396 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df     F p-value
## s(DiaPres):Treatment 1.500  1.500 7.671 0.00683 **
## s(height):Treatment  8.323  9.086 2.776 0.00454 **
## s(weight)            5.655  6.815 1.995 0.07008 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 31/32
## R-sq.(adj) =  0.366   Deviance explained =   49%
## GCV = 21.703  Scale est. = 17.296    n = 91
```

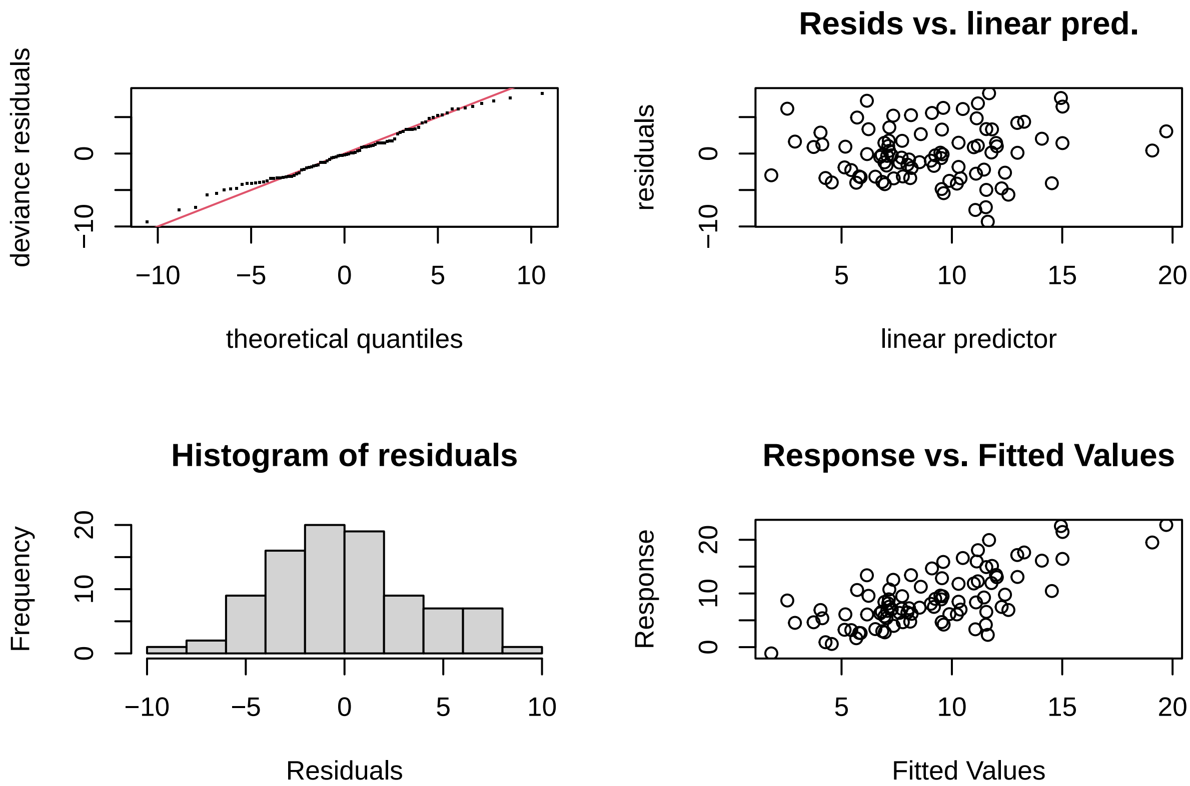
```r
plot(gam_4.1,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= c
```




s(height,8.32):Treatment
60
40
20
0
−20
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height




```
gam.check(gam_4.1)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 17 iterations.
## The RMS GCV score gradient at convergence was 8.491236e-07 .
## The Hessian was positive definite.
## Model rank =  31 / 32
##
## 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(DiaPres):Treatment 10.00  1.50    1.01    0.48
## s(height):Treatment  10.00  8.32    0.93    0.22
## s(weight)             9.00  5.65    1.08    0.78
```

Here we tried a new approach and combined variables in a smoothing function. Also we removed DiaPress which is reasonable as it is linear correlated looking at the correlation plot at the beginning of the file.

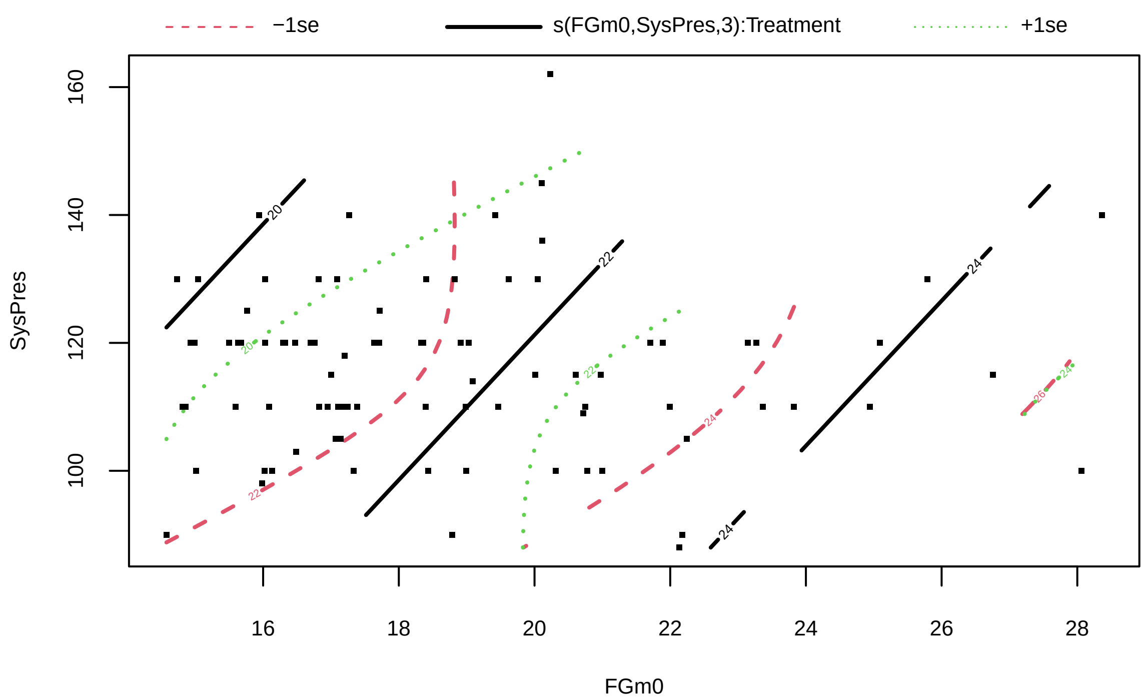
```r
gam_5.1 <- gam(FGm12 ~ SysPres +  s(FGm0,SysPres, by=Treatment, k=5) + s(height, k=25, bs="cr") + s(wei
summary(gam_5.1)
```

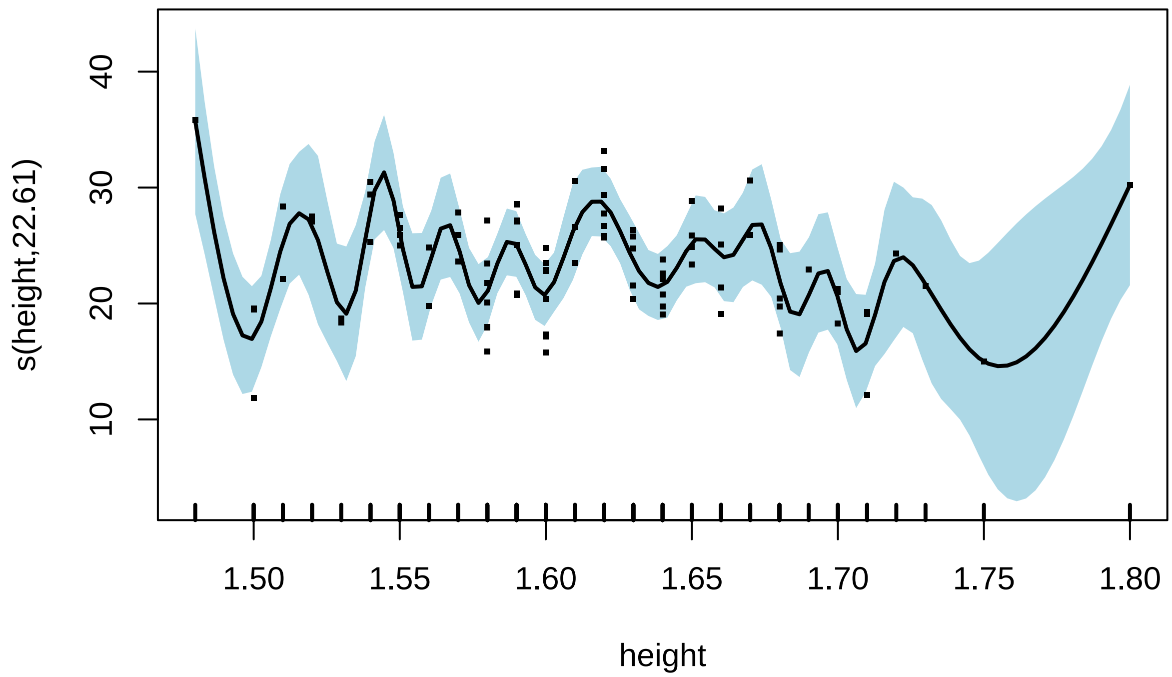
```
##
## Family: gaussian
## Link function: identity
##
```

```
## Formula:
## FGm12 ~ SysPres + s(FGm0, SysPres, by = Treatment, k = 5) + s(height,
##     k = 25, bs = "cr") + s(weight, fx = TRUE, bs = "cr")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23.5257     6.9751   3.373  0.00138 **
## SysPres      -0.1011     0.0608  -1.662  0.10222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                              edf Ref.df      F  p-value
## s(FGm0,SysPres):Treatment  3.00    3.0 12.624 2.72e-06 ***
## s(height)                 22.61   23.7  3.479 0.000137 ***
## s(weight)                  9.00    9.0  2.082 0.047355 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.568   Deviance explained = 73.9%
## GCV = 19.718  Scale est. = 11.785    n = 91
```

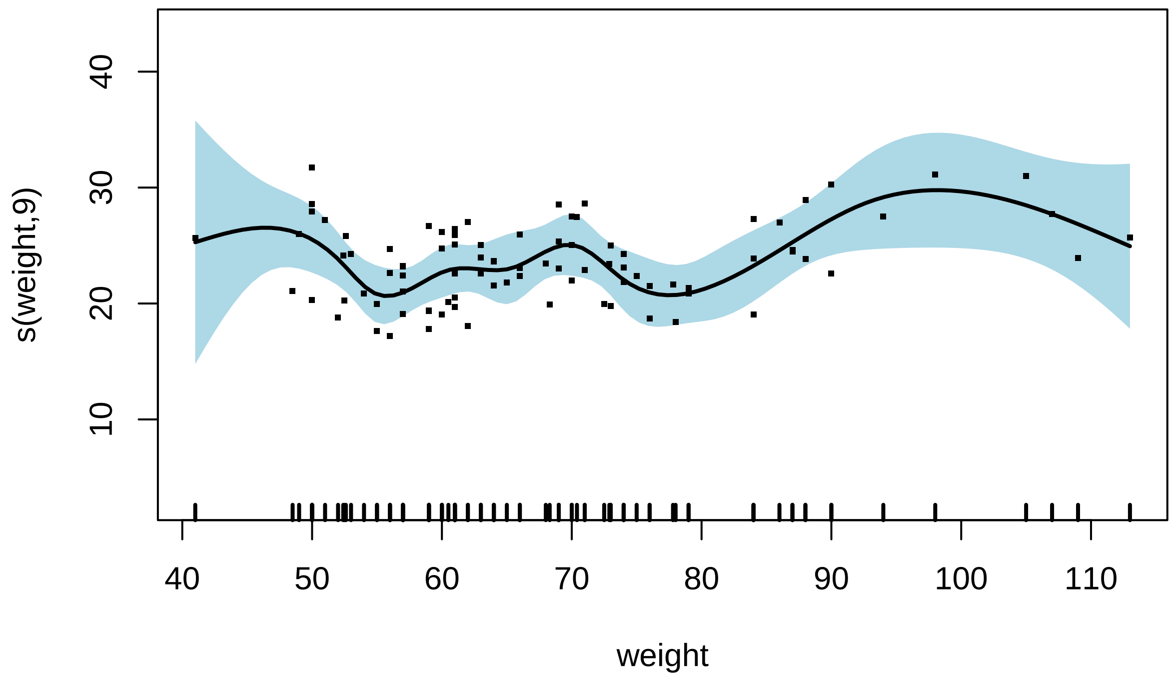
Since the interaction is taking into account the different factors from Treatment we had a lot of uncertainty

```r
plot(gam_5.1,residuals=TRUE, shade=TRUE, shade.col="lightblue", seWithMean=TRUE, cex=3, lwd=2, shift= c
```



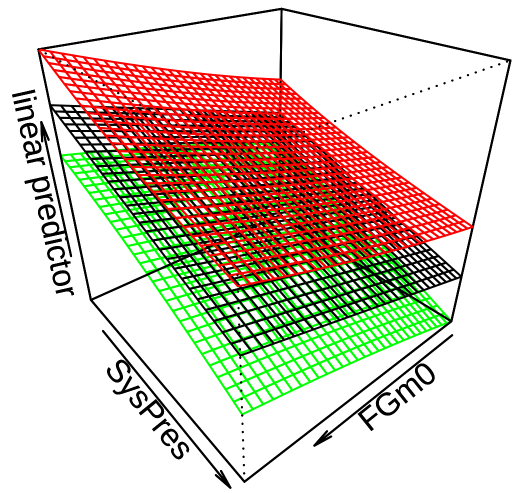


10
20
30
40
s(height,22.61)
1.50
1.55
1.60
1.65
1.70
1.75
1.80
height



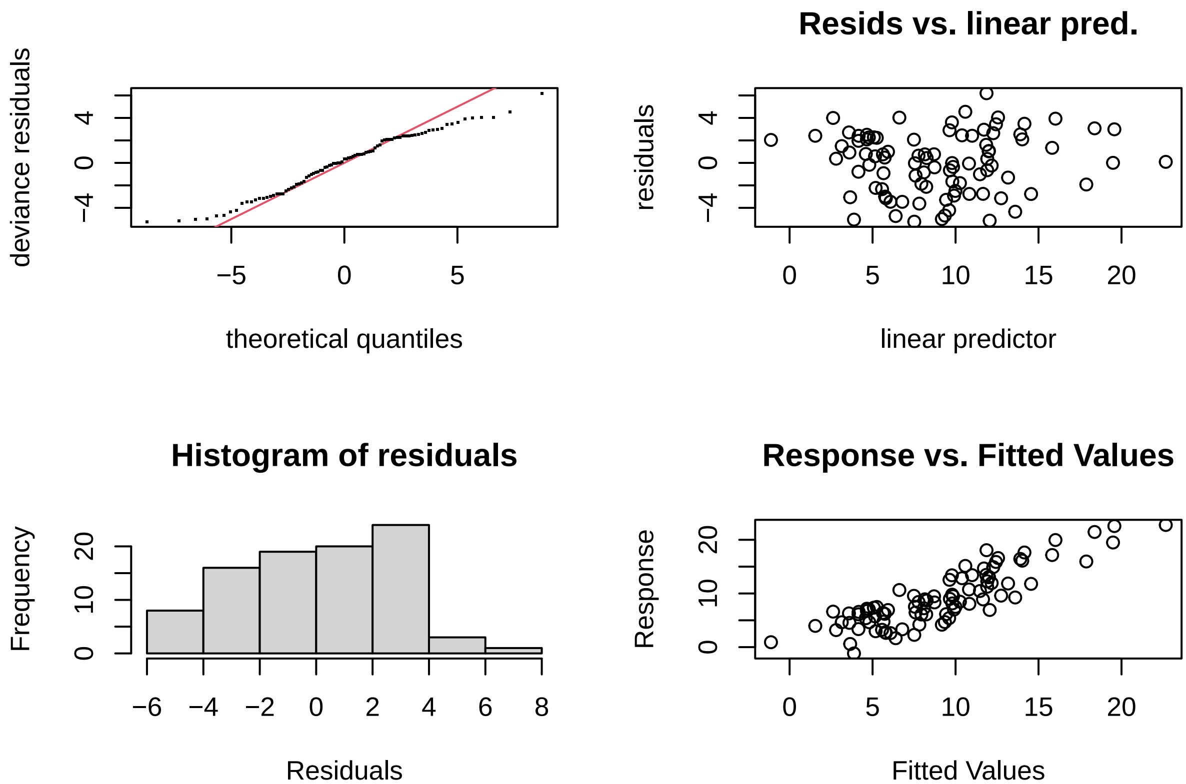

This model has a comparably high value for deviance explained and the partial plots are not constant as for many previous models. R-sq is also higher.

```
vis.gam(gam_5.1, view = c("FGm0","SysPres"), plot.type = "persp", se=2,
            theta = 140,  phi=30)
```




red/green are +/− 2 s.e.

```
gam.check(gam_5.1)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 15 iterations.
## The RMS GCV score gradient at convergence was 2.314776e-06 .
## The Hessian was positive definite.
## Model rank =  40 / 40
##
## 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(FGm0,SysPres):Treatment  5.0  3.0    1.13    0.92
## s(height)                 24.0 22.6    1.10    0.82
## s(weight)                  9.0  9.0    1.16    0.90
```

These latest plots look better than the previous ones. However, the Q-Q plot is not straight line, and the histogram lacks a well-defined bell shape, implying that the model isn't perfect.

Looking at the console result, the model converged without issues, and the p-values from the smooths are considerably high. This suggests that we have a good number of basis functions (k) to be confident in our model

## Anova tests

```r
anova(gam_0.1,gam_1.1,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ FGm0 + SysPres + DiaPres + height + weight + Treatment
## Model 2: FGm12 ~ s(FGm0, by = Treatment) + s(SysPres, by = Treatment) +
##     s(DiaPres, by = Treatment) + s(height, by = Treatment) +
##     s(weight, by = Treatment)
##   Resid. Df Resid. Dev     Df Deviance      F   Pr(>F)
## 1    84.000     1985.3
## 2    71.426     1310.8 12.574   674.47 2.9991 0.001657 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gam_1.1 better

```r
anova(gam_0.1,gam_1.2,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ FGm0 + SysPres + DiaPres + height + weight + Treatment
## Model 2: FGm12 ~ s(FGm0) + s(SysPres) + s(DiaPres) + s(height) + s(weight)
##   Resid. Df Resid. Dev     Df Deviance     F Pr(>F)
## 1    84.000     1985.3
## 2    76.273     1745.7 7.7274   239.53 1.391  0.216
```

Gam_1.2 better

```r
anova(gam_1.2,gam_1.1,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ s(FGm0) + s(SysPres) + s(DiaPres) + s(height) + s(weight)
## Model 2: FGm12 ~ s(FGm0, by = Treatment) + s(SysPres, by = Treatment) +
##     s(DiaPres, by = Treatment) + s(height, by = Treatment) +
##     s(weight, by = Treatment)
##   Resid. Df Resid. Dev     Df Deviance      F    Pr(>F)
## 1    76.273     1745.7
## 2    71.426     1310.8 4.8466   434.93 5.0175 0.0006115 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gam_1.1 better (with treatment)

```r
anova(gam_2.1,gam_2.2,test="F")
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: FGm12 ~ s(FGm0, by = Treatment) + SysPres + s(DiaPres, by = Treatment) +
##     s(height, by = Treatment) + s(weight, by = Treatment)
## Model 2: FGm12 ~ FGm0 + s(SysPres, by = Treatment) + s(DiaPres, by = Treatment) +
##     s(height, by = Treatment) + s(weight, by = Treatment)
##   Resid. Df Resid. Dev     Df Deviance     F  Pr(>F)
## 1    76.073     1492.0
## 2    71.230     1292.5 4.8424   199.43 2.328 0.05313 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gam_2.2 better

```r
anova(gam_2.2,gam_2.3,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ FGm0 + s(SysPres, by = Treatment) + s(DiaPres, by = Treatment) +
##     s(height, by = Treatment) + s(weight, by = Treatment)
## Model 2: FGm12 ~ s(FGm0, by = Treatment) + s(SysPres, by = Treatment) +
##     DiaPres + s(height, by = Treatment) + s(weight, by = Treatment)
##   Resid. Df Resid. Dev       Df Deviance      F Pr(>F)
## 1    71.230     1292.5
## 2    71.537     1307.1 -0.30648   -14.53 2.6798 0.1114
```

Gam_2.2 better

```r
anova(gam_2.2,gam_1.1,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ FGm0 + s(SysPres, by = Treatment) + s(DiaPres, by = Treatment) +
##     s(height, by = Treatment) + s(weight, by = Treatment)
## Model 2: FGm12 ~ s(FGm0, by = Treatment) + s(SysPres, by = Treatment) +
##     s(DiaPres, by = Treatment) + s(height, by = Treatment) +
##     s(weight, by = Treatment)
##   Resid. Df Resid. Dev       Df Deviance     F  Pr(>F)
## 1    71.230     1292.5
## 2    71.426     1310.8 -0.19567  -18.263 5.276 0.05659 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gam_1.1 better

```r
anova(gam_3.1,gam_3.2,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ FGm0 + SysPres + s(DiaPres, by = Treatment) + s(height,
##     by = Treatment) + s(weight, by = Treatment)
## Model 2: FGm12 ~ FGm0 + s(SysPres, by = Treatment) + DiaPres + s(height,
##     by = Treatment) + s(weight, by = Treatment)
##   Resid. Df Resid. Dev       Df Deviance      F Pr(>F)
## 1    70.879     1272.0
## 2    71.374     1297.3 -0.49486  -25.354 2.9296 0.1056
```

Gam_3.1 better

```r
anova(gam_3.1,gam_1.1,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ FGm0 + SysPres + s(DiaPres, by = Treatment) + s(height,
##     by = Treatment) + s(weight, by = Treatment)
## Model 2: FGm12 ~ s(FGm0, by = Treatment) + s(SysPres, by = Treatment) +
##     s(DiaPres, by = Treatment) + s(height, by = Treatment) +
##     s(weight, by = Treatment)
##   Resid. Df Resid. Dev       Df Deviance      F  Pr(>F)
## 1    70.879     1272.0
## 2    71.426     1310.8 -0.54728  -38.819 4.0557 0.06707 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gam_3.1 better

```r
anova(gam_3.1,gam_4.1,test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ FGm0 + SysPres + s(DiaPres, by = Treatment) + s(height,
##     by = Treatment) + s(weight, by = Treatment)
## Model 2: FGm12 ~ FGm0 + SysPres + s(DiaPres, by = Treatment) + s(height,
##     by = Treatment) + s(weight)
##   Resid. Df Resid. Dev      Df Deviance     F  Pr(>F)
## 1    70.879     1272.0
## 2    70.599     1254.3 0.27947   17.657 3.653 0.08299 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gam_4.1 better

```r
anova(gam_4.1, gam_5.1,test="F")
```

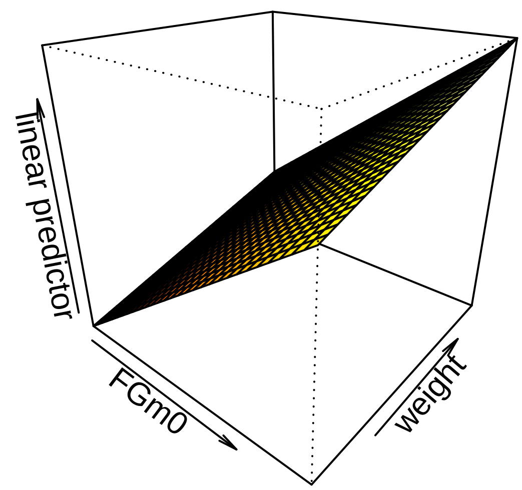
```
## Analysis of Deviance Table
##
## Model 1: FGm12 ~ FGm0 + SysPres + s(DiaPres, by = Treatment) + s(height,
##     by = Treatment) + s(weight)
## Model 2: FGm12 ~ SysPres + s(FGm0, SysPres, by = Treatment, k = 5) + s(height,
##     k = 25, bs = "cr") + s(weight, fx = TRUE, bs = "cr")
##   Resid. Df Resid. Dev     Df Deviance      F   Pr(>F)
## 1    70.599     1254.3
## 2    53.302      641.0 17.297   613.33 3.0087 0.001051 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gam_5.1 better

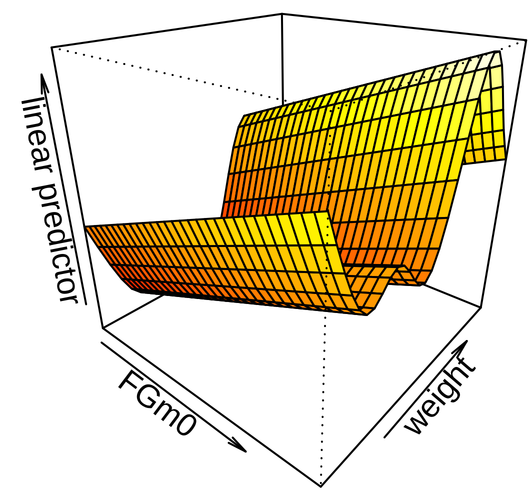
Comparing the models with anova, the model gam_5.1 is the best obtained model out of them.

### Compare plot of a "bad" and a "good" model

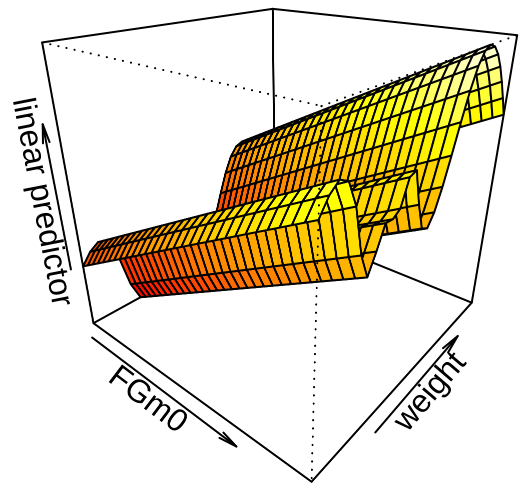
```r
vis.gam(gam_0.1,view=c("FGm0","weight"),
        theta = 40, phi = 25, r = sqrt(3), d = 1)
```




```r
vis.gam(gam_3.1,view=c("FGm0","weight"),
        theta = 40, phi = 25, r = sqrt(3), d = 1)
```




```r
vis.gam(gam_5.1,view=c("FGm0","weight"),
        theta = 40, phi = 25, r = sqrt(3), d = 1)
```




We can see that the better performing model is more complex. The complexity might be necessary to represent the relation of the variables to our prediction value properly.