Exercise 8

Nikolaus Czernin

```
# library("MASS")
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# install.packages("ROCit")
library("ROCit")
library("knitr")
# install.packages("glmnet")
library("glmnet")
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
library(mgcv)
```

```
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
##     collapse
##
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
set.seed(11721138)
```

Loading & Preprocessing

```
data <- Diabetes
# ?Diabetes

data <- data %>%
  mutate(dtest = ifelse(dtest == "+", 1, 0)) %>%
  select(dtest, everything(),
         -id,
         -waist,
         -hip,
         -height,
         -weight,
         -hdl,
         -chol,
         -time.ppn) %>%
  # select(dtest, bmi, whr, ratio, stab.glu, age, glyhb) %>%
  na.omit()
```

I dropped some variables:

- id because it is just an identifier for the row
- whr is perfectly dependent on two other variables, hip and waist, which i can therefore remove
- bmi for the same reason, it is calculated from weight and height, which is remove
- ratio for the same reason, it includes the information of cholesterol and hdl, which i remove

```
train.idx <- sample(1:nrow(data), nrow(data)%/%4*3)
train <- data[train.idx, ]
test <- data[-train.idx, ]
```

1. Logistic Regression

```
model.lr <- glm(dtest ~., family=binomial, data=train)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
model.lr
```

```
##
## Call:  glm(formula = dtest ~ ., family = binomial, data = train)
##
## Coefficients:
##    (Intercept)        stab.glu           ratio           glyhb  locationLouisa
##      -259.2786          0.2278         -5.0049         30.0880         33.5310
##            age      gendermale     framemedium      framesmall           bp.1s
##        -0.1456        -24.2008        -12.4295         -8.5109          1.3964
##          bp.1d           bp.2s           bp.2d             bmi             whr
##        -0.4513         -1.2910          0.8982         -1.4778         41.2899
##
## Degrees of Freedom: 95 Total (i.e. Null);  81 Residual
## Null Deviance:       110.1
## Residual Deviance: 4.684e-09     AIC: 30
```

```
yhat <- predict(model.lr, newdata = test, type="response") %>% print()
```

```
##           16           34           50           61           69           92
## 2.220446e-16 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e+00
##          100          102          114          119          131          144
## 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 9.999984e-01
##          154          159          170          189          201          204
## 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16
##          230          236          271          285          293          299
## 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16
##          322          332          347          351          353          354
## 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16
##          358          359          373          374          390
## 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e-16
```

```
predicted <- yhat %>% round()
observed <-test$dtest
cm <- table(predicted, observed) %>% print()
```

```
##          observed
## predicted  0  1
##         0 30  0
##         1  0  5
```

```
MCR <- sum(cm[1, 2], cm[2, 1]) / sum(cm)
print(paste("Misclassification rate:", MCR %>% round(4)))
```

```
## [1] "Misclassification rate: 0"
```

When fitting the model I got a warning that the model did not converge and that fitted probabilities of 0 and 1 occurred. This signifies that there is some crass overfitting or strong dependencies between predictors.

2. Sparse logistic regression

```r
x <- train[,-1] %>% makeX()
y <- train[,1]
model.slr <- cv.glmnet(x, y, family = "binomial")

yhat <- predict(model.slr, newx = test[,-1] %>% makeX(), type="response")
predicted <- yhat %>% round()
observed <-test$dtest
cm <- table(predicted, observed) %>% print()
```

```
##          observed
## predicted  0  1
##         0 30  0
##         1  0  5
```

```r
MCR <- sum(cm[1, 2], cm[2, 1]) / sum(cm)
print(paste("Misclassification rate:", MCR %>% round(4)))
```

```
## [1] "Misclassification rate: 0"
```

Now we have no misclassifications at all.

2. Generalized additive models

a

```r
m1 <- gam(dtest ~
            s(ratio) +
            s(stab.glu) +
            s(glyhb) +
            s(age) +
            s(bp.1s) +
            s(bp.1d) +
            s(bp.2s) +
            s(bp.2d) +
            s(bmi) +
            location +
            gender +
            frame +
            s(whr)
          , data=train, family="binomial")
```

By selecting the compound variables bmi and whr instead of their components (weight & height for the bmi for example) I avoided having to limit the degrees of freedom of the smoothing splines.

c

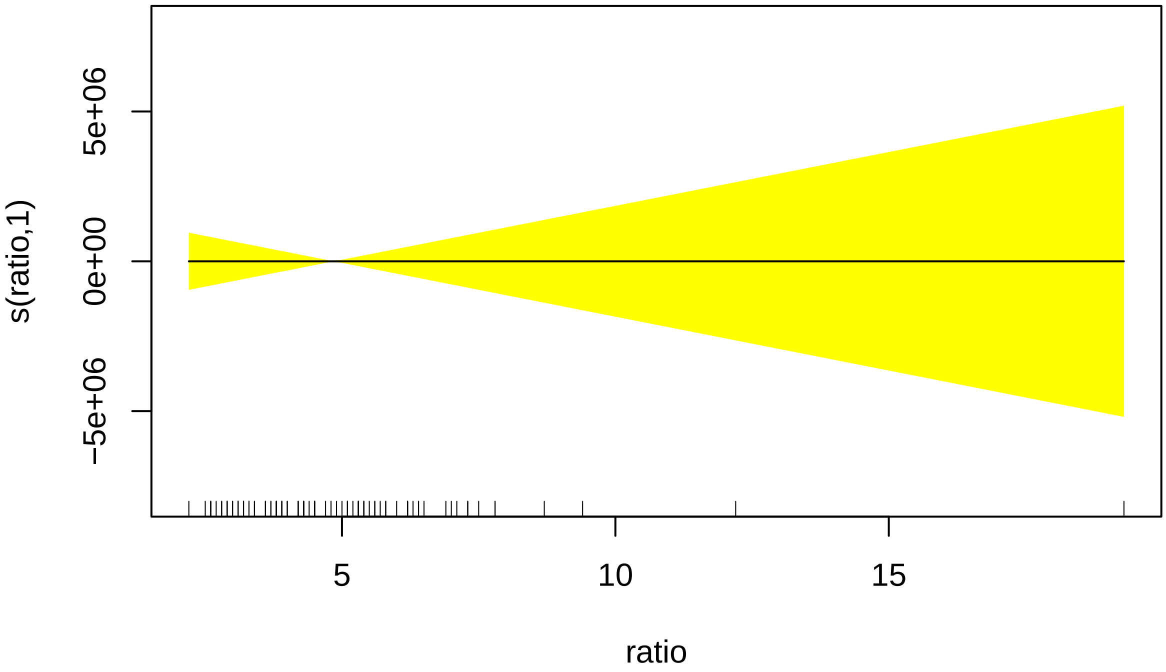
```r
m1 %>% summary()
```

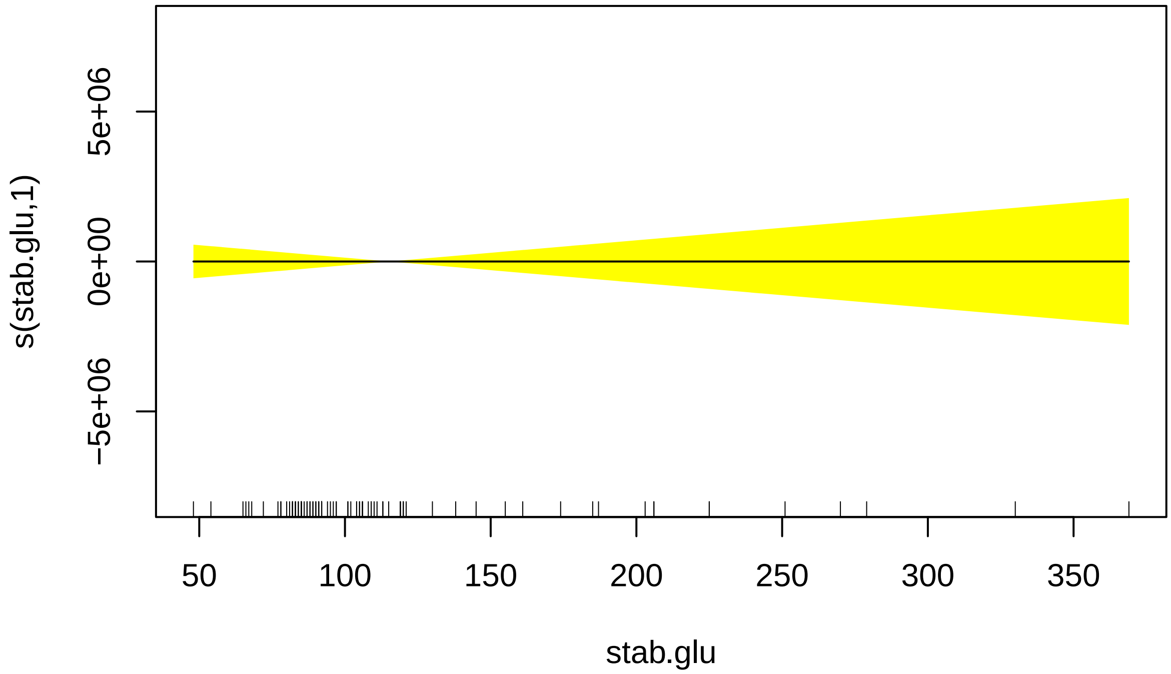
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(ratio) + s(stab.glu) + s(glyhb) + s(age) + s(bp.1s) +
##     s(bp.1d) + s(bp.2s) + s(bp.2d) + s(bmi) + location + gender +
##     frame + s(whr)
##
## Parametric coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -30.17  787896.36       0        1
## locationLouisa      39.09  584472.95       0        1
## gendermale         -29.12  627742.14       0        1
## framemedium        -14.04  426139.36       0        1
## framesmall         -10.49 1007433.90       0        1
##
## Approximate significance of smooth terms:
##              edf Ref.df Chi.sq p-value
## s(ratio)       1      1      0     0.5
## s(stab.glu)    1      1      0     0.5
## s(glyhb)       1      1      0     0.5
## s(age)         1      1      0     0.5
## s(bp.1s)       1      1      0     0.5
## s(bp.1d)       1      1      0     0.5
## s(bp.2s)       1      1      0     0.5
## s(bp.2d)       1      1      0     0.5
## s(bmi)         1      1      0     0.5
## s(whr)         1      1      0     1.0
##
## R-sq.(adj) =      1   Deviance explained =  100%
## UBRE = -0.6875  Scale est. = 1         n = 96
```

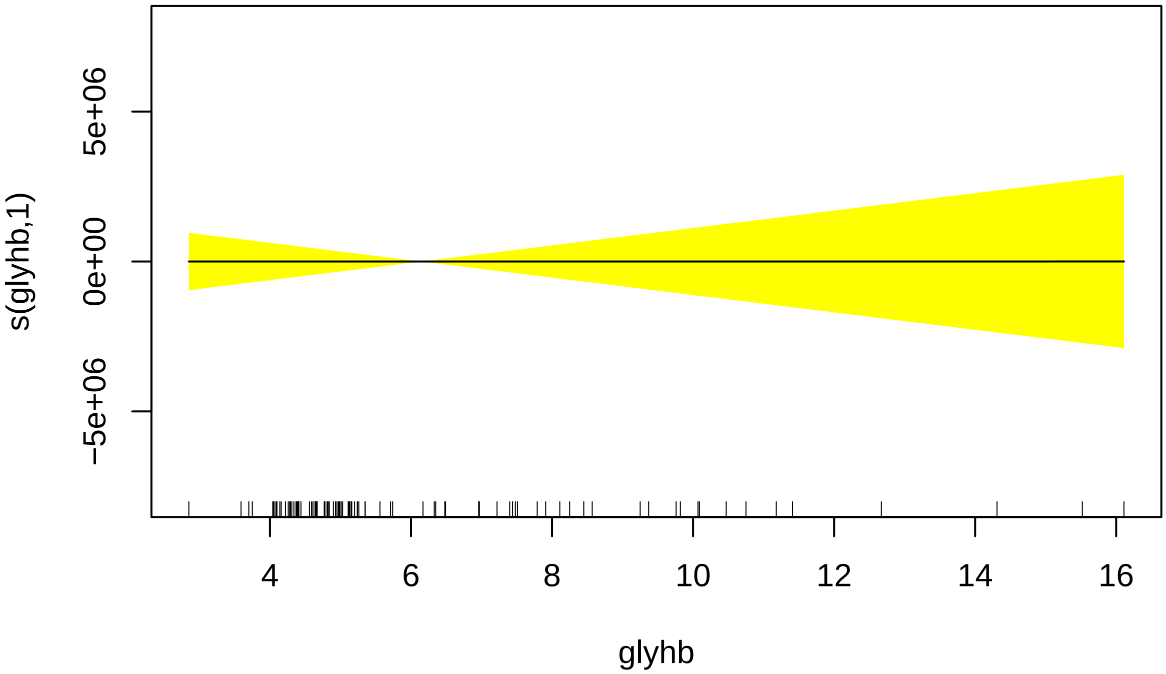
None of the computed computed smoothing splines seem to be significant though. From looking at the plots printed below and the estimated degrees of freedom we can see that the splines are all linear, which kind of defeats the purpose of using GAMs in the first place.

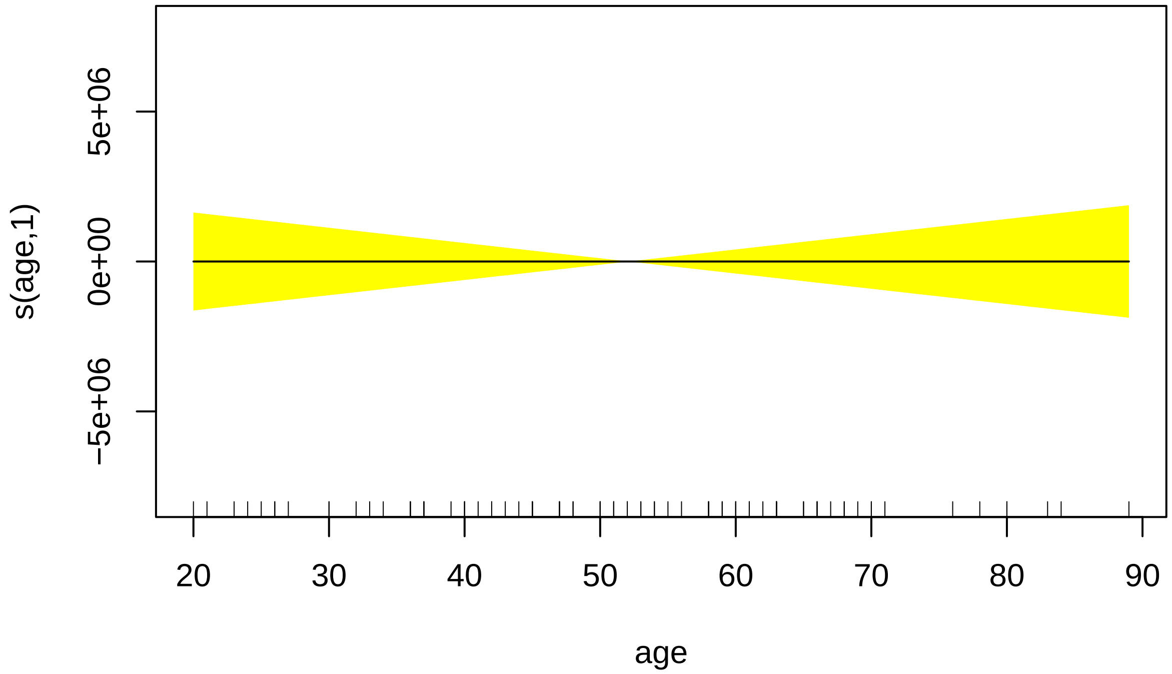
d

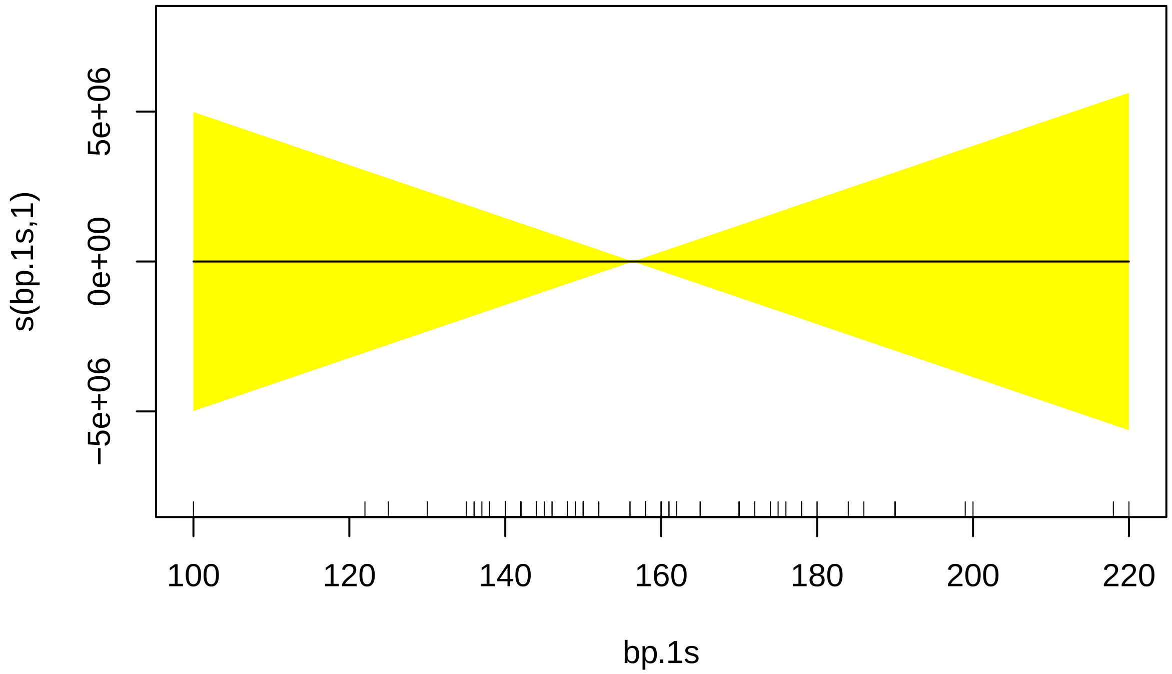
```r
m1 %>% plot(shade=TRUE,shade.col="yellow")
```

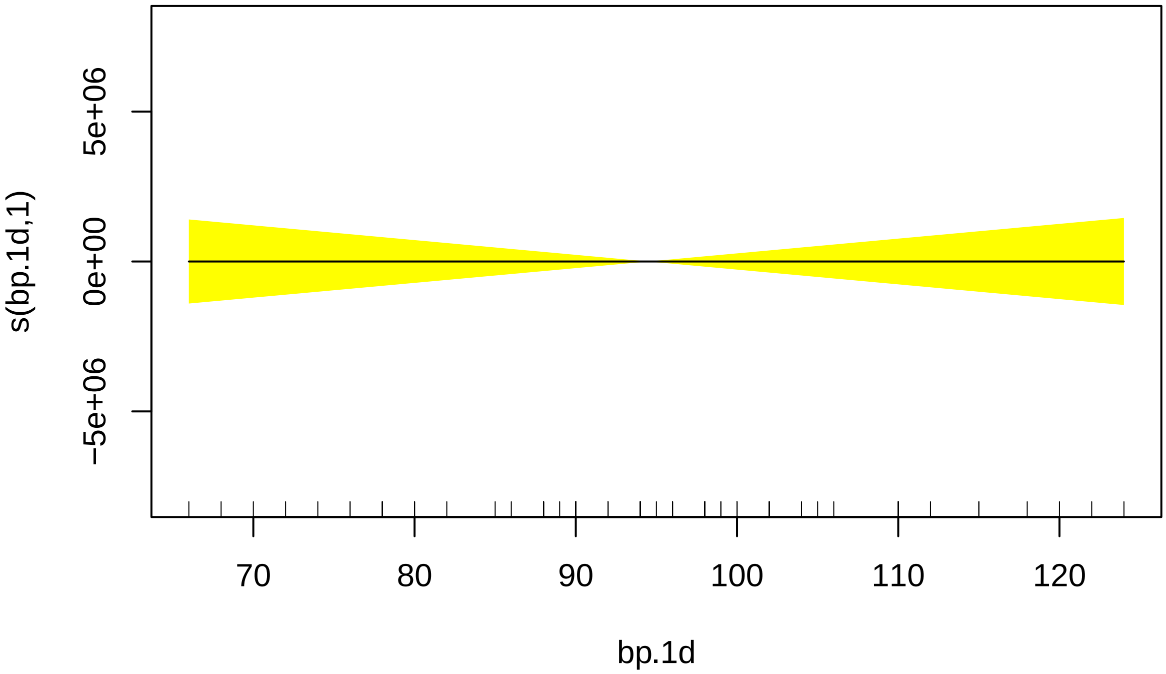

s(ratio,1)
5e+06
0e+00
−5e+06
5
10
15
ratio

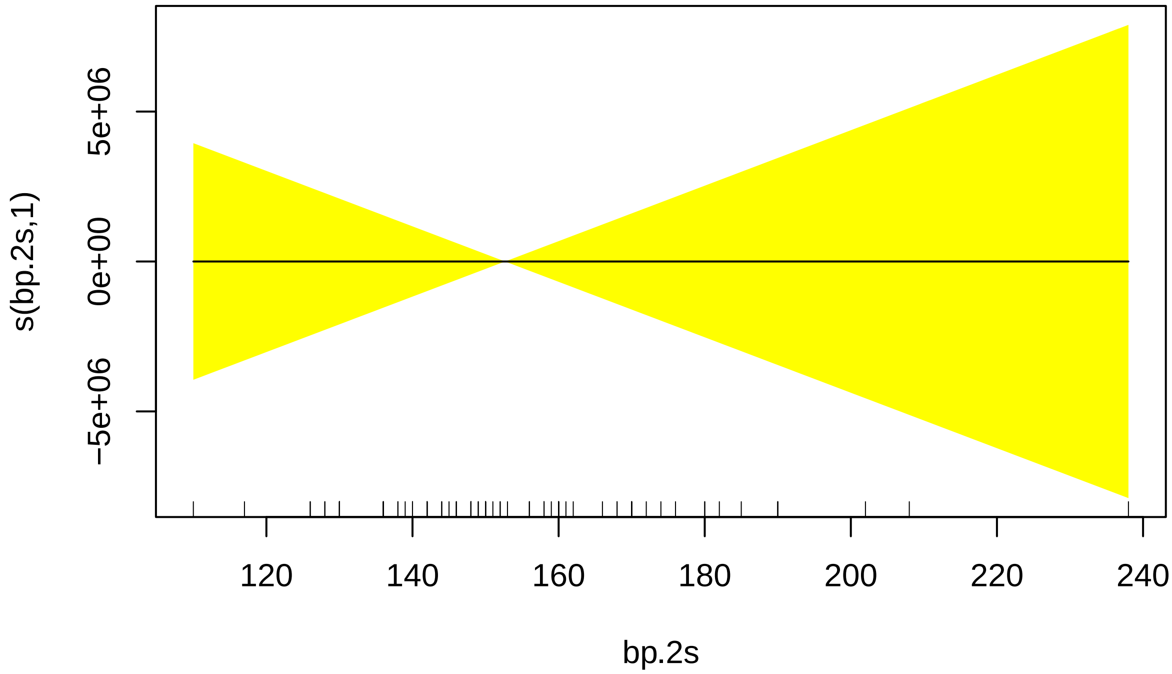

s(stab.glu,1)
5e+06
0e+00
−5e+06
50
100
150
200
250
300
350
stab.glu

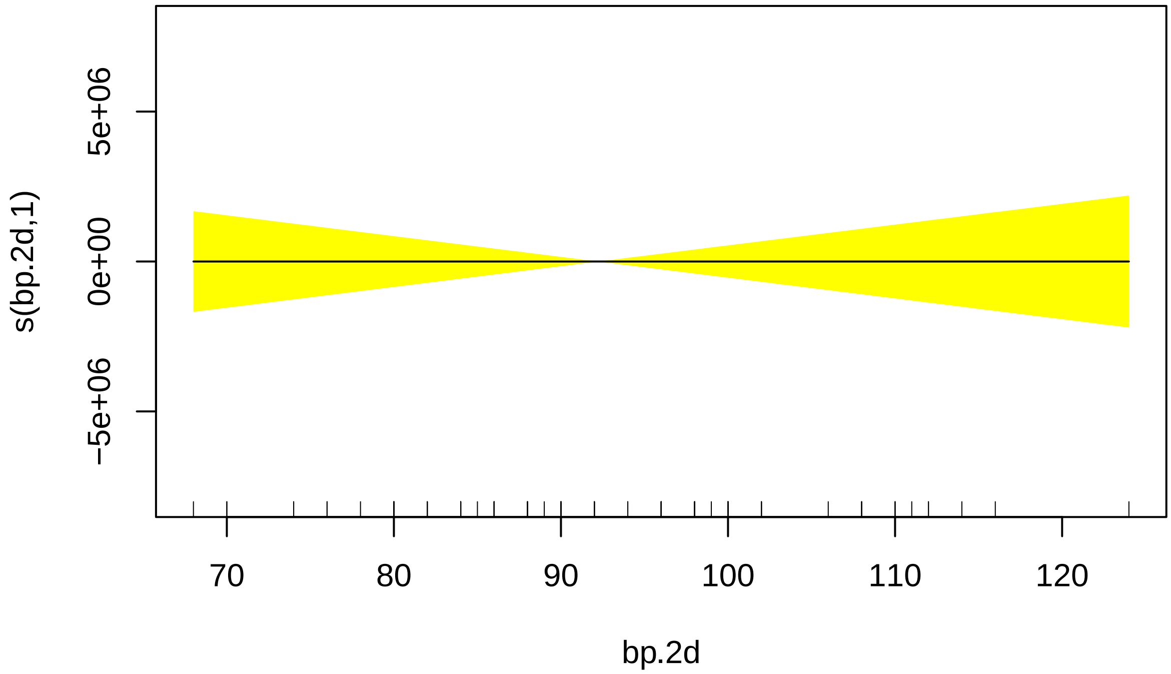

s(glyhb,1)
5e+06
0e+00
−5e+06
4
6
8
10
12
14
16
glyhb

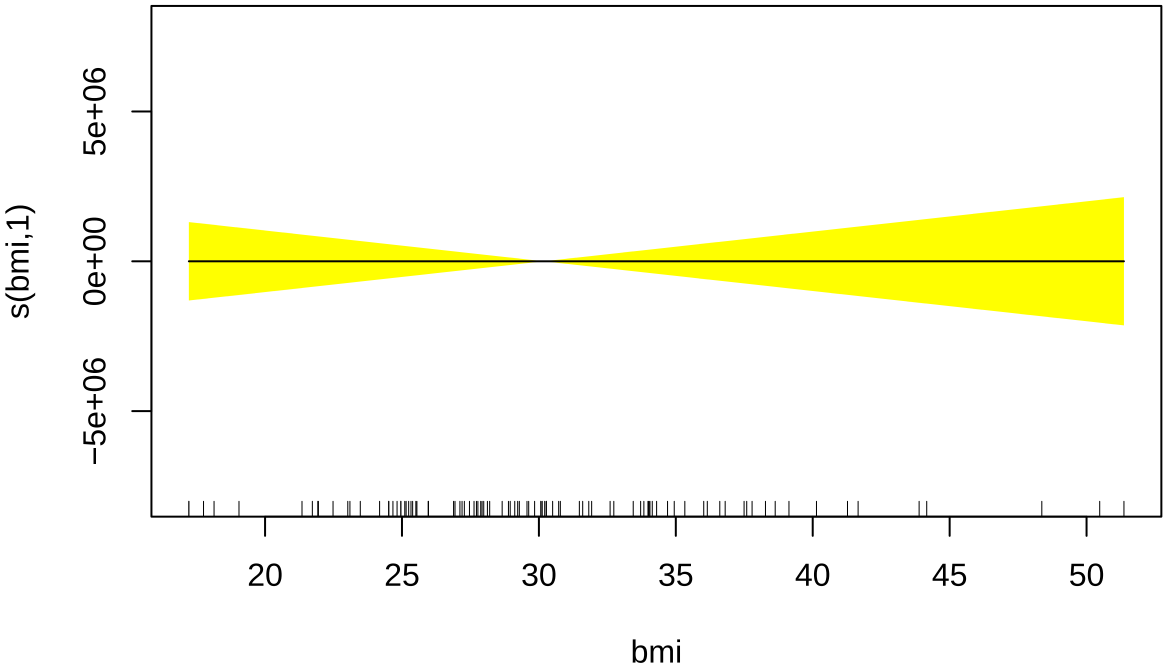

5e+06
0e+00
−5e+06
s(age,1)
20
30
40
50
60
70
80
90
age

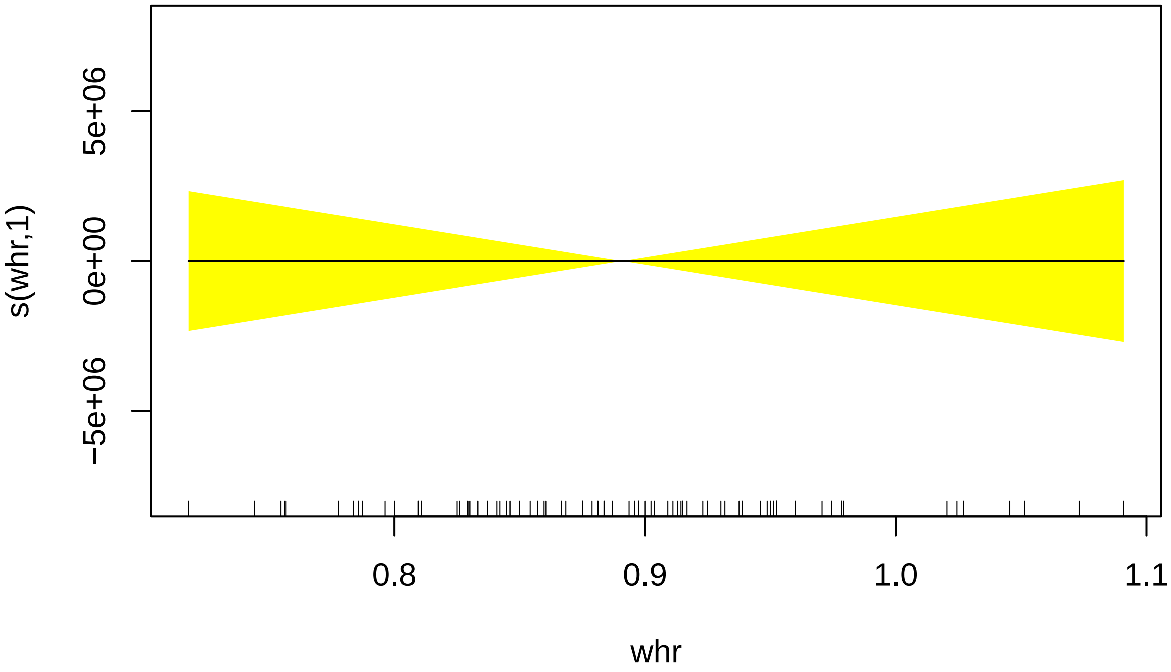

5e+06
0e+00
−5e+06
s(bp.1s,1)
100
120
140
160
180
200
220
bp.1s


5e+06
0e+00
−5e+06
s(bp.1d,1)
70
80
90
100
110
120
bp.1d


s(bp.2s,1)
5e+06
0e+00
−5e+06
120
140
160
180
200
220
240
bp.2s


5e+06
0e+00
−5e+06
s(bp.2d,1)
70
80
90
100
110
120
bp.2d


5e+06
0e+00
−5e+06
s(bmi,1)
20
25
30
35
40
45
50
bmi



e

```
yhat <- predict(m1, se.fit=TRUE, test[,-1], type="response")

predicted <- yhat %>% .$fit %>% round()
observed <-test$dtest
cm <- table(predicted, observed) %>% print()
```

```
##          observed
## predicted  0  1
##         0 30  0
##         1  0  5
```

```
MCR <- sum(cm[1, 2], cm[2, 1]) / sum(cm)
print(paste("Misclassification rate:", MCR %>% round(4)))
```

```
## [1] "Misclassification rate: 0"
```

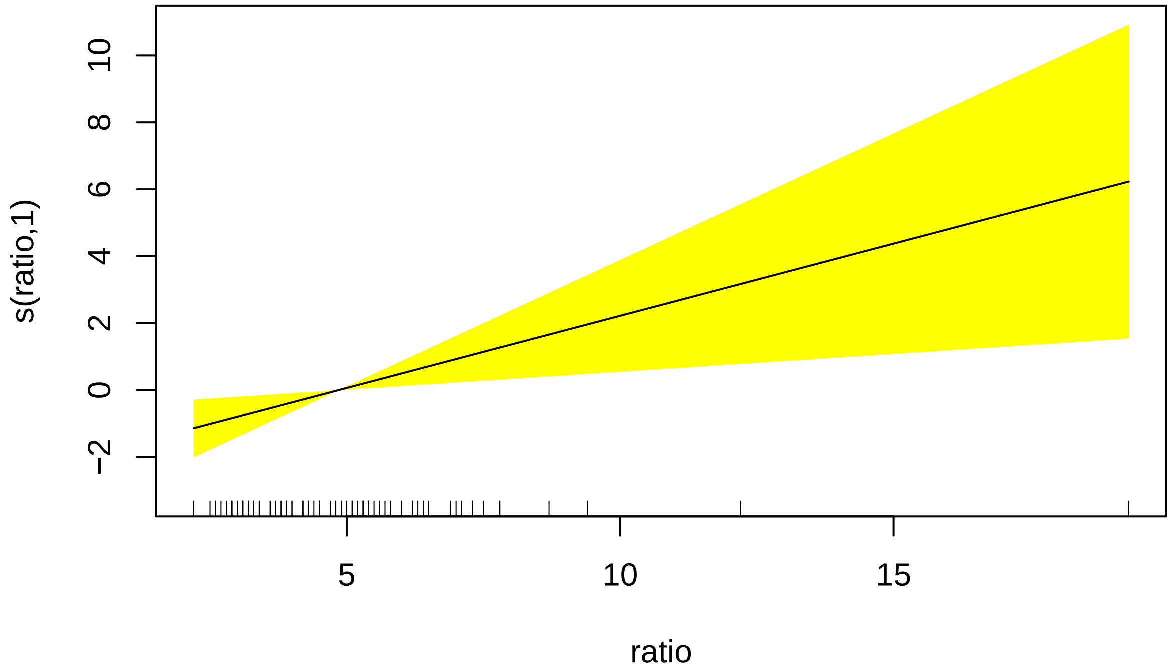
```
?step.gam
```

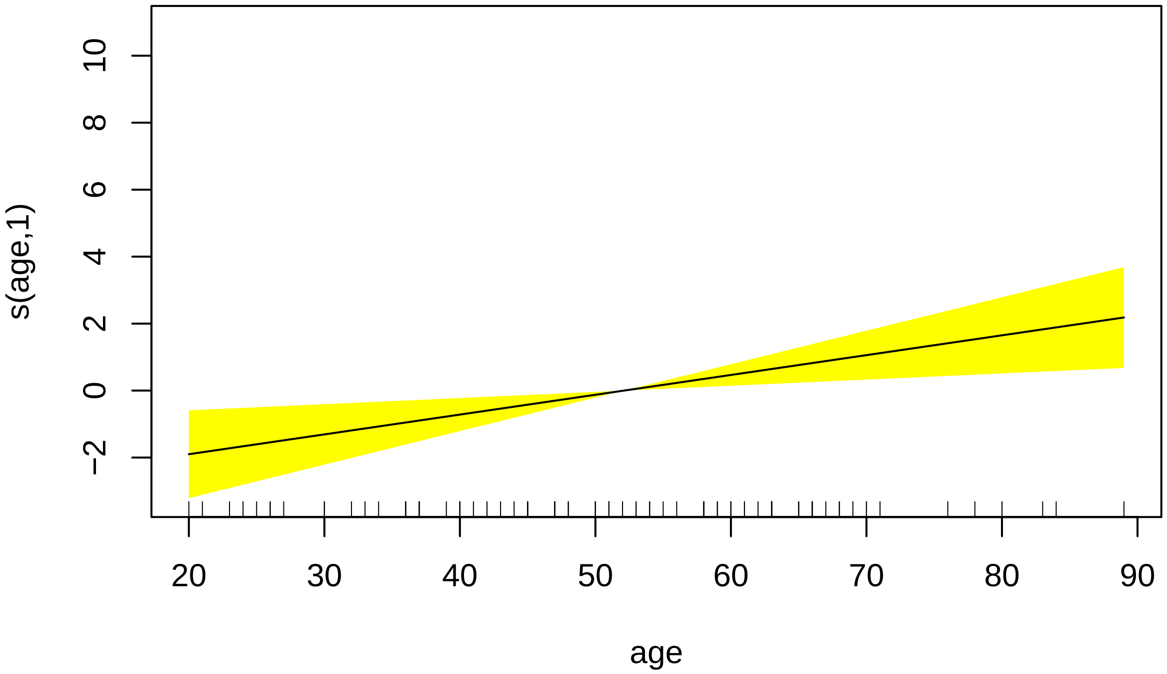
e2: Fitting with fewer variables

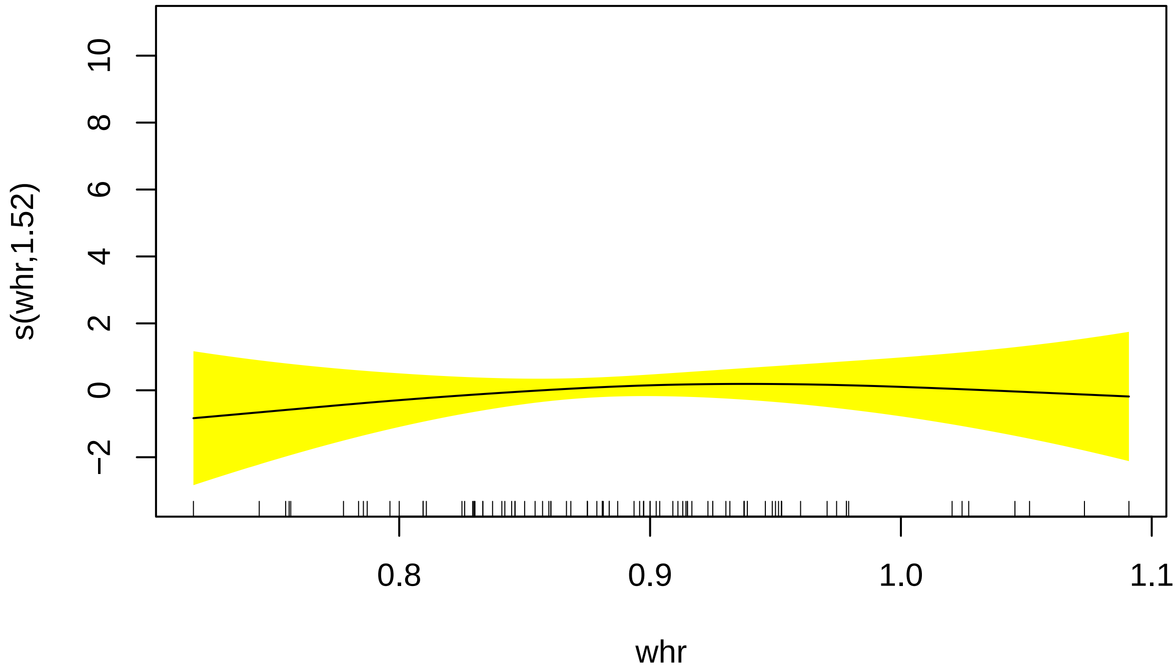
```
m2 <- gam(dtest ~
            s(ratio) +
            # s(stab.glu) +
            # s(glyhb) +
            s(age) +
            # s(bp.1s) +
            # s(bp.1d) +
            # s(bp.2s) +
            # s(bp.2d) +
            # s(bmi) +
            # location +
            # gender +
            # frame +
            s(whr)
          , data=train, family="binomial")
m2 %>% summary()
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(ratio) + s(age) + s(whr)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3312     0.2963  -4.492 7.05e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##            edf Ref.df Chi.sq p-value
## s(ratio) 1.000  1.000  7.057 0.00790 **
## s(age)   1.000  1.000  8.408 0.00374 **
## s(whr)   1.518  1.886  1.074 0.61552
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.187   Deviance explained =    21%
## UBRE = 0.00023281  Scale est. = 1         n = 96
```

```
m2 %>% plot(shade=TRUE,shade.col="yellow")
```


10
8
6
4
2
0
−2
s(ratio,1)
5
10
15
ratio


s(age,1)
−2
0
2
4
6
8
10
20
30
40
50
60
70
80
90
age



```
yhat <- predict(m2, se.fit=TRUE, test[,-1], type="response")
predicted <- yhat %>% .$fit %>% round()
observed <-test$dtest
cm <- table(predicted, observed) %>% print()
```

```
##          observed
## predicted  0  1
##         0 30  3
##         1  0  2
```

```
MCR <- sum(cm[1, 2], cm[2, 1]) / sum(cm)
print(paste("Misclassification rate:", MCR %>% round(4)))
```

```
## [1] "Misclassification rate: 0.0857"
```

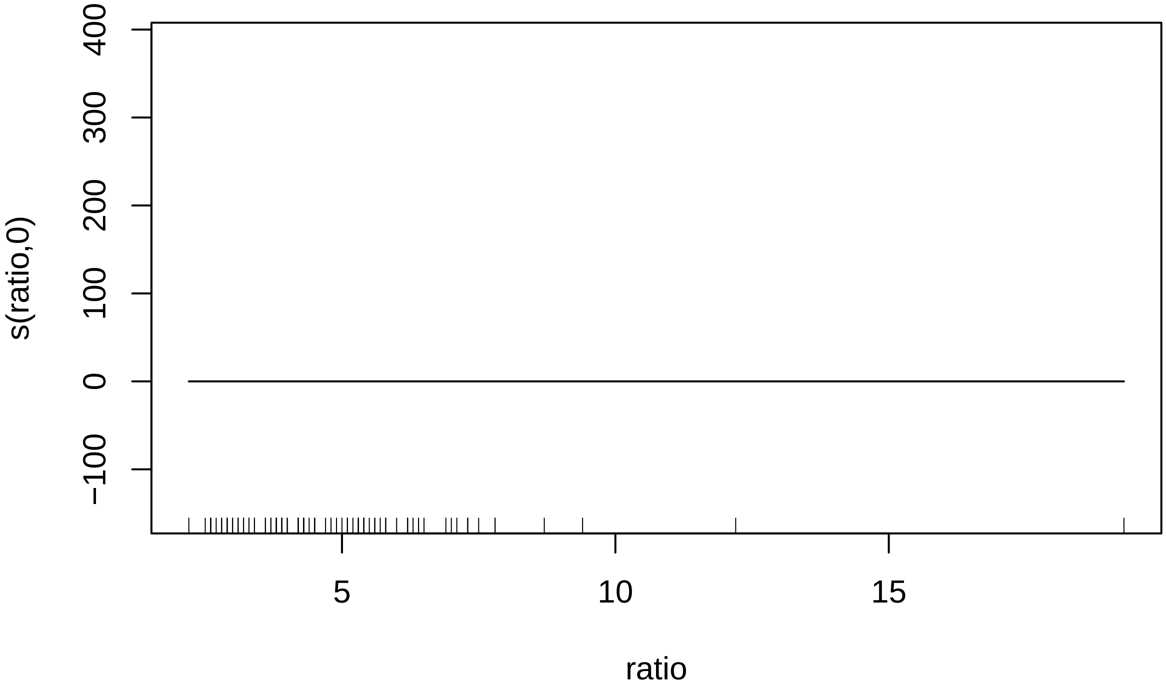
If I select fewer variables, they end up being significant, some of them even not just linear. The misclassification rate in turn also goes up a little, but still remains super low.

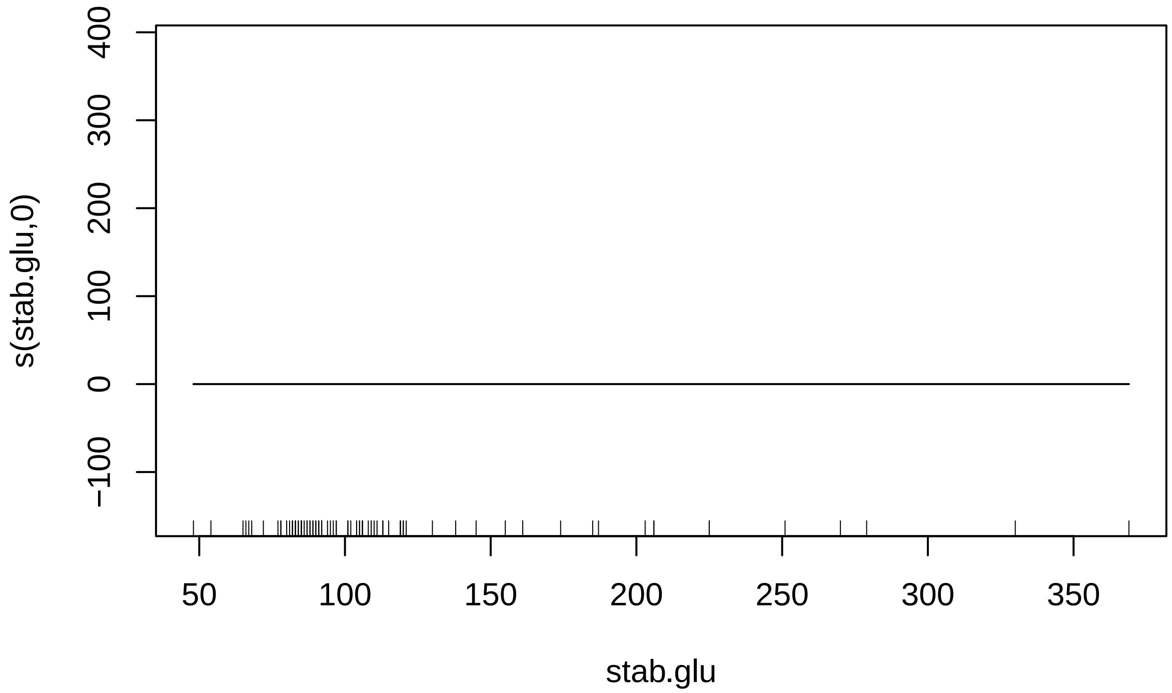
f: modelling via step.gam

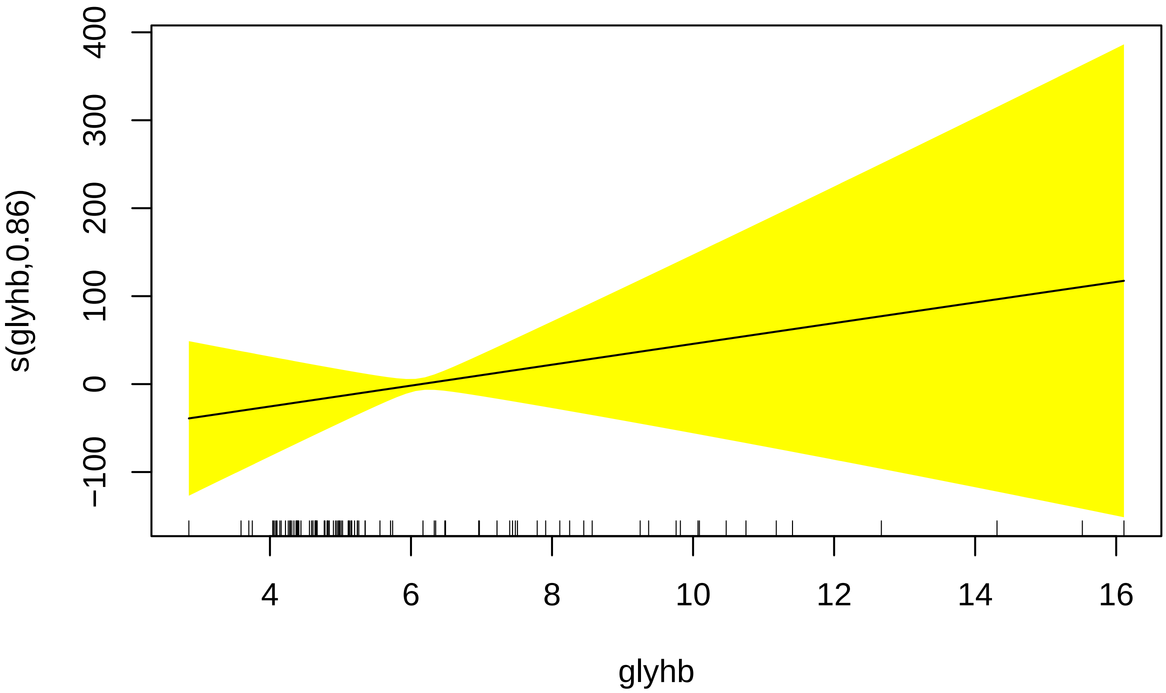
```
m3 <- gam(dtest ~
            s(ratio,bs="ts") +
            s(stab.glu,bs="ts") +
            s(glyhb,bs="ts") +
            s(age) +
            s(bp.1s,bs="ts") +
            s(bp.1d,bs="ts") +
            s(bp.2s,bs="ts") +
            s(bp.2d,bs="ts") +
            s(bmi,bs="ts") +
            location +
            gender +
            frame +
            s(whr,bs="ts")
          , data=train, family="binomial")
m3 %>% summary()
```

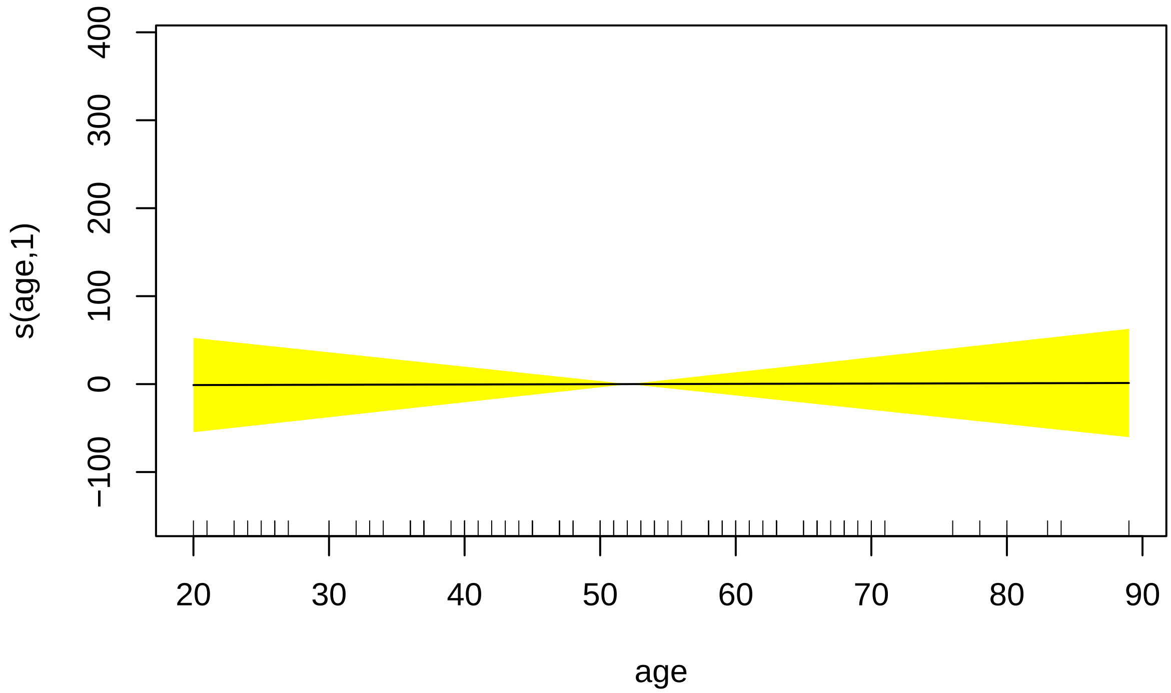
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(ratio, bs = "ts") + s(stab.glu, bs = "ts") + s(glyhb,
##     bs = "ts") + s(age) + s(bp.1s, bs = "ts") + s(bp.1d, bs = "ts") +
##     s(bp.2s, bs = "ts") + s(bp.2d, bs = "ts") + s(bmi, bs = "ts") +
##     location + gender + frame + s(whr, bs = "ts")
##
## Parametric coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -8.470     21.446  -0.395    0.693
## locationLouisa    9.093     21.103   0.431    0.667
## gendermale       -7.356     20.137  -0.365    0.715
## framemedium      -9.467     23.506  -0.403    0.687
## framesmall      -10.600   4799.921  -0.002    0.998
##
## Approximate significance of smooth terms:
##                   edf Ref.df Chi.sq p-value
## s(ratio)    6.743e-06      8  0.000   0.920
## s(stab.glu) 1.748e-06      8  0.000   0.930
## s(glyhb)    8.622e-01      8  0.817   0.330
## s(age)      1.000e+00      1  0.002   0.967
## s(bp.1s)    4.765e-07      9  0.000   0.917
## s(bp.1d)    6.373e-07      9  0.000   0.956
## s(bp.2s)    8.836e-07      9  0.000   0.933
## s(bp.2d)    2.733e-07      9  0.000   0.917
## s(bmi)      2.801e-07      9  0.000   0.993
## s(whr)      1.483e-06      9  0.000   0.988
##
## R-sq.(adj) =      1   Deviance explained =   100%
## UBRE = -0.85665  Scale est. = 1         n = 96
```

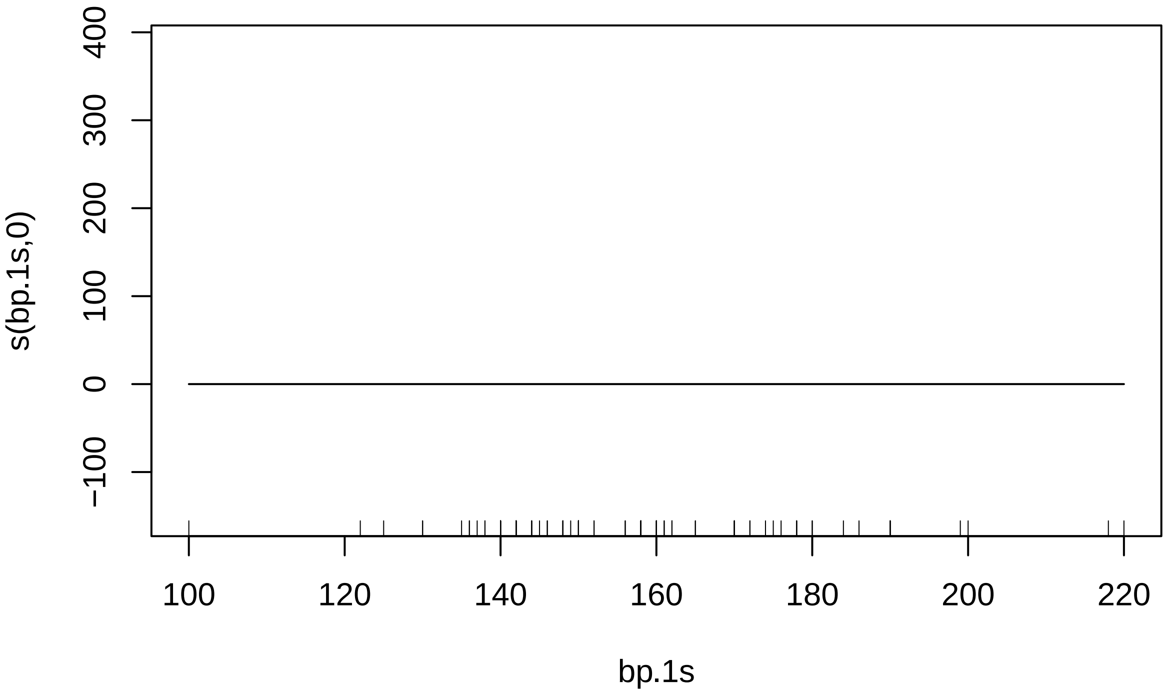
```r
m3 %>% plot(shade=TRUE,shade.col="yellow")
```

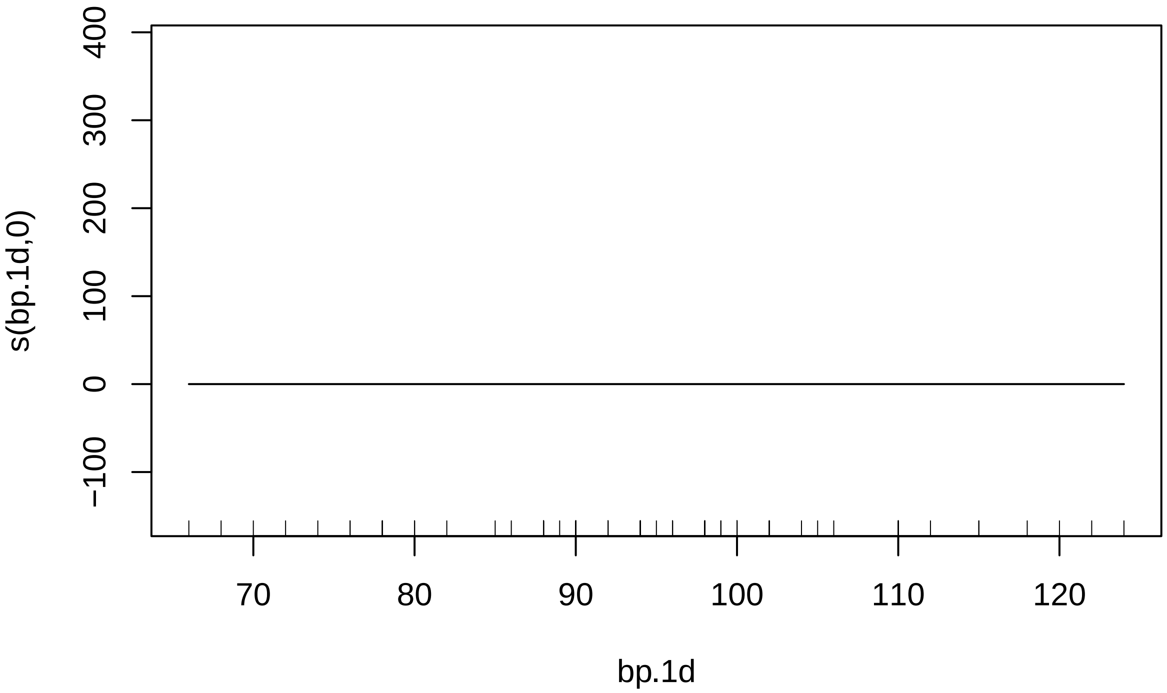


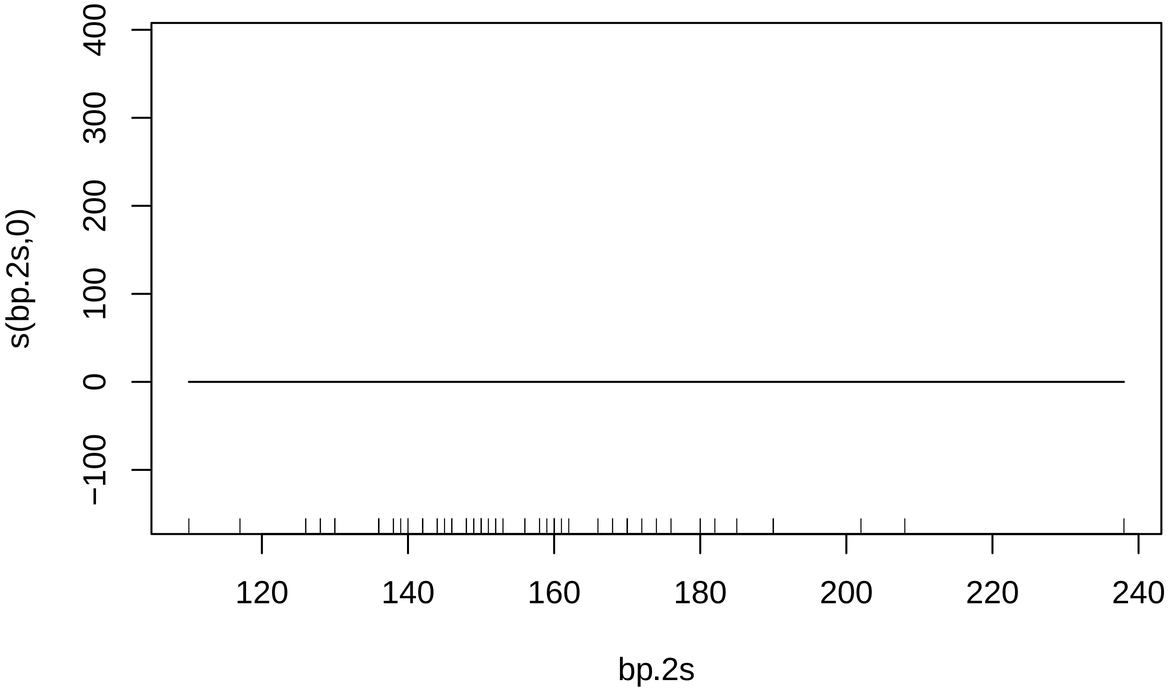

s(stab.glu,0)
400
300
200
100
0
−100
50
100
150
200
250
300
350
stab.glu

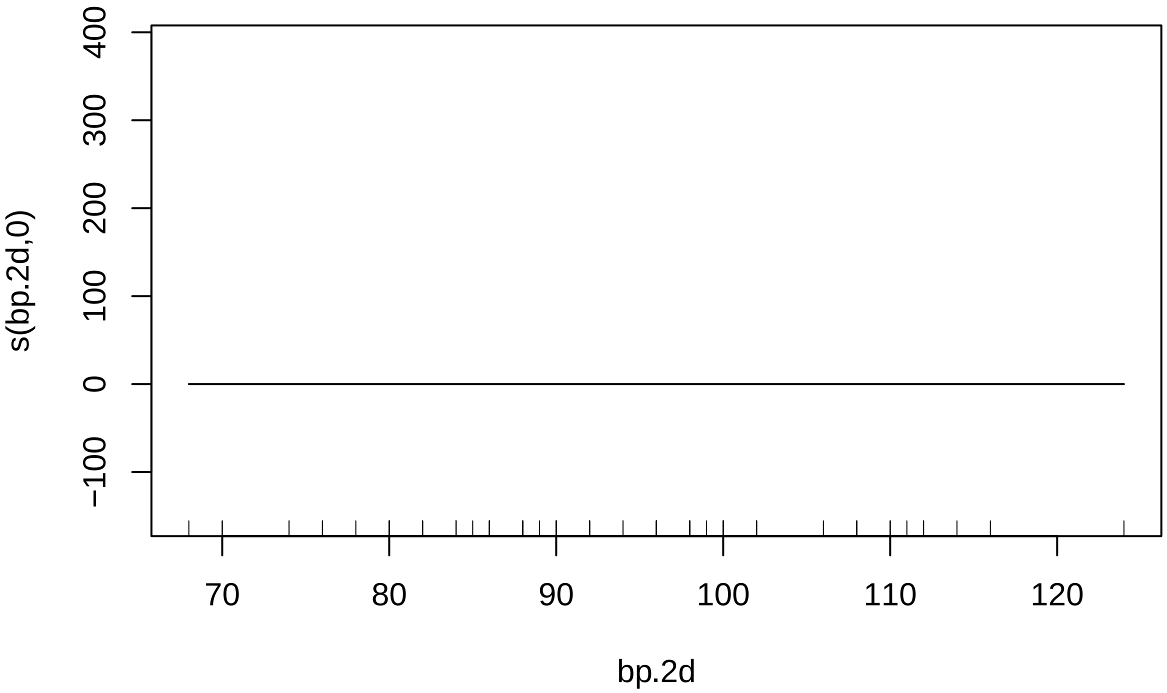

s(glyhb,0.86)
400
300
200
100
0
−100
4
6
8
10
12
14
16
glyhb

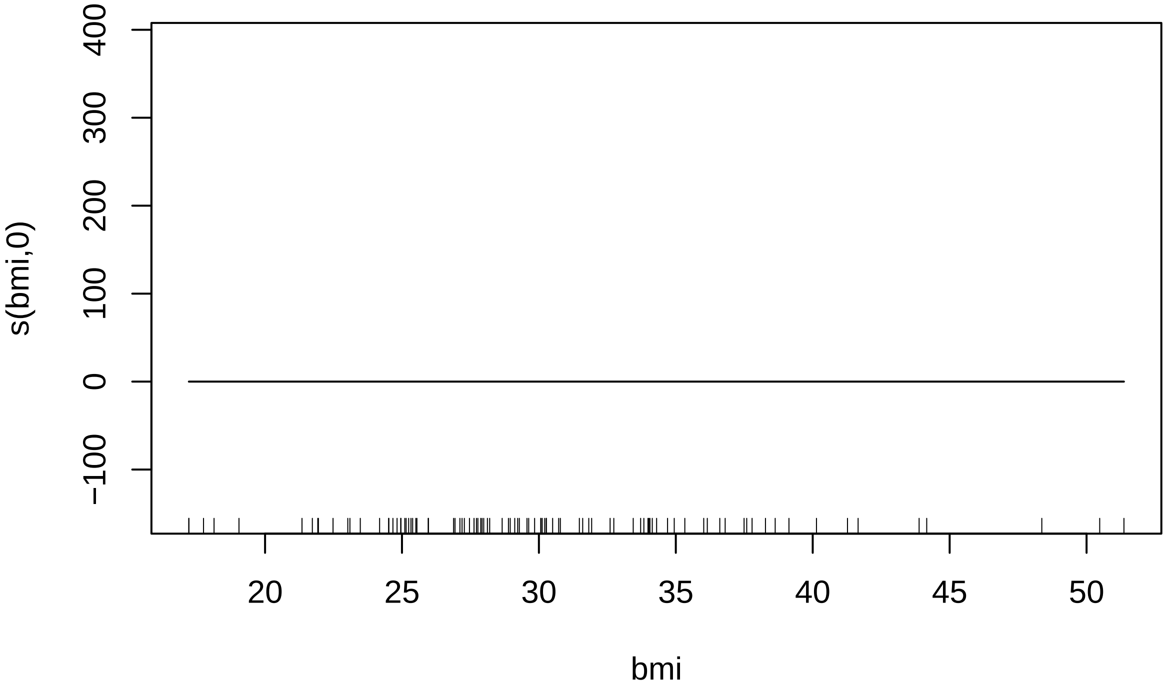

s(age,1)
400
300
200
100
0
−100
20
30
40
50
60
70
80
90
age

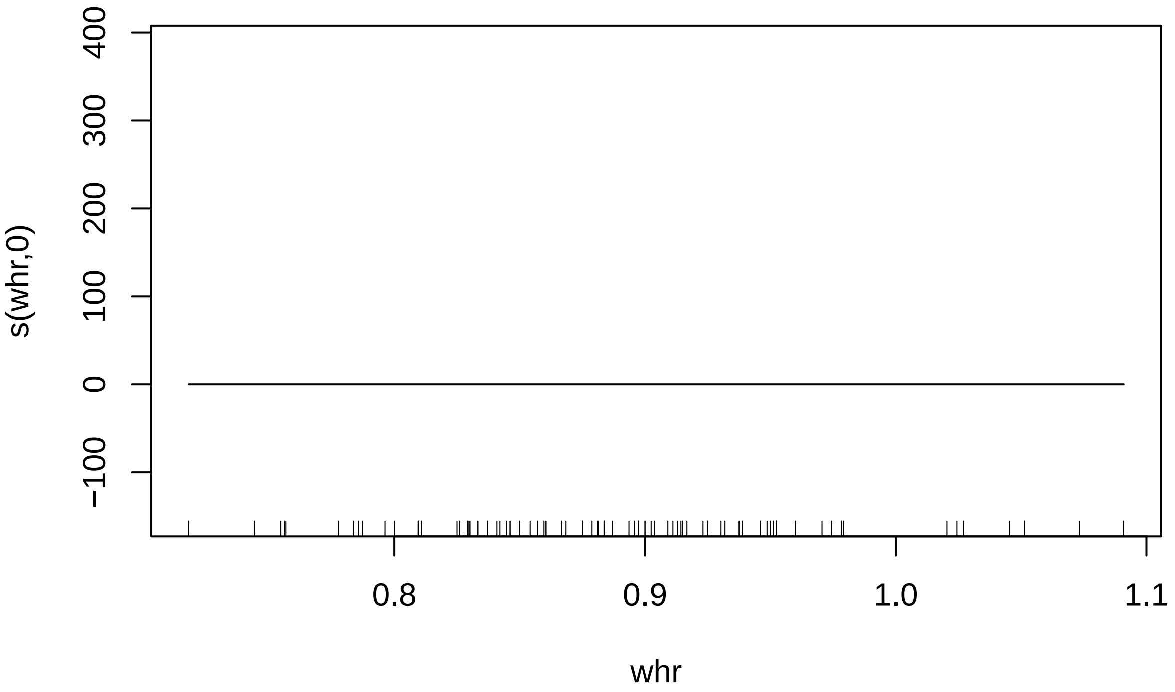

s(bp.1s,0)
400
300
200
100
0
−100
100
120
140
160
180
200
220
bp.1s


400
300
200
100
0
−100
s(bp.1d,0)
70
80
90
100
110
120
bp.1d


s(bp.2s,0)
400
300
200
100
0
−100
120
140
160
180
200
220
240
bp.2s


s(bp.2d,0)
400
300
200
100
0
−100
70
80
90
100
110
120
bp.2d


s(bmi,0)
400
300
200
100
0
−100
20
25
30
35
40
45
50
bmi



```r
yhat <- predict(m3, se.fit=TRUE, test[,-1], type="response")
predicted <- yhat %>% .$fit %>% round()
observed <-test$dtest
cm <- table(predicted, observed) %>% print()
```

```
##          observed
## predicted  0  1
##         0 30  0
##         1  0  5
```

```r
MCR <- sum(cm[1, 2], cm[2, 1]) / sum(cm)
print(paste("Misclassification rate:", MCR %>% round(4)))
```

```
## [1] "Misclassification rate: 0"
```

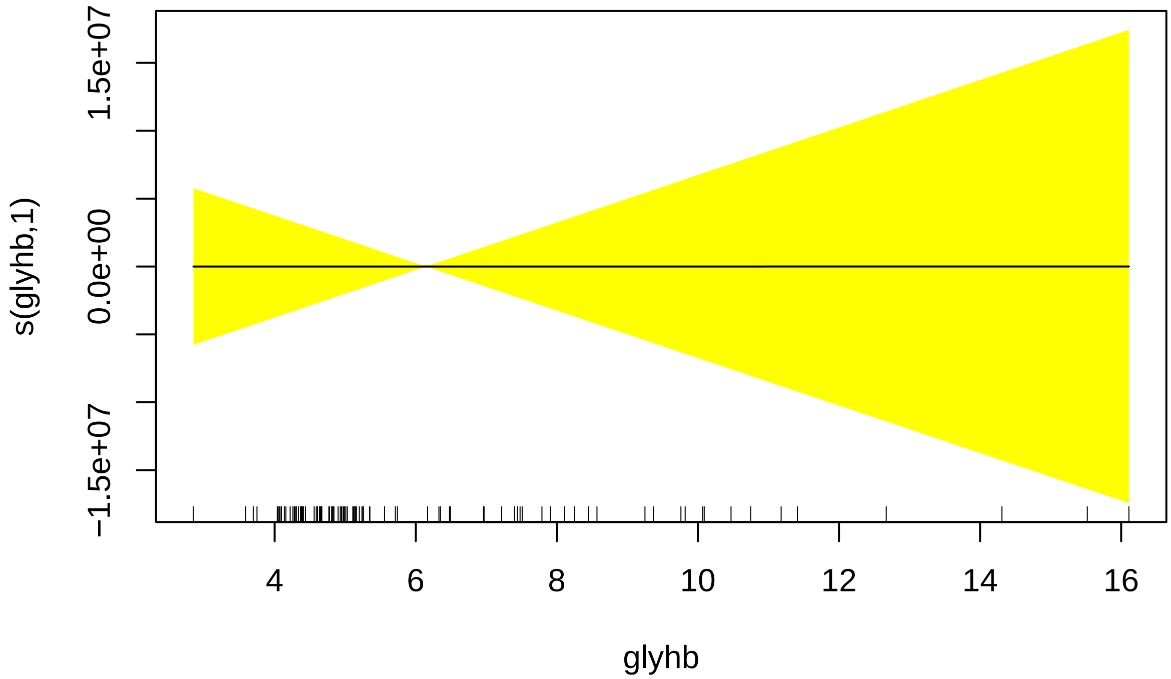
When using the thin splate regression spline smoother, most variables have estimated degrees of freedom of nearly zero, merely glyhb and age seem to have been “selected”.
The model still has a misclassification rate of 0 though. When using the cubic regression splines, the same variables are selected.

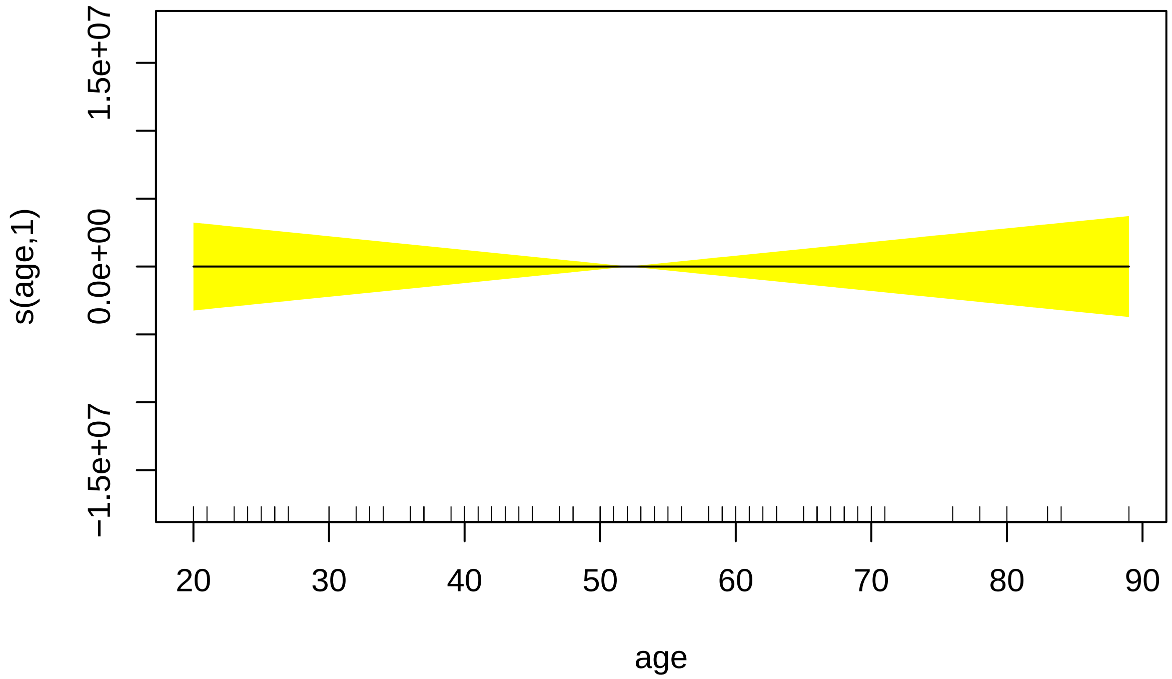
g: Fitting with the variables selected by “step.gam”

```
m5 <- gam(dtest ~
            s(glyhb) +
            s(age)
          , data=train, family="binomial")
m5 %>% summary()
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(glyhb) + s(age)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -184     848422       0        1
##
## Approximate significance of smooth terms:
##          edf Ref.df Chi.sq p-value
## s(glyhb)   1      1      0     1.0
## s(age)     1      1      0     0.5
##
## R-sq.(adj) =      1   Deviance explained =  100%
## UBRE = -0.9375  Scale est. = 1         n = 96
```

```
m5 %>% plot(shade=TRUE,shade.col="yellow")
```


s(glyhb,1)
1.5e+07
0.0e+00
−1.5e+07
4
6
8
10
12
14
16
glyhb



```
yhat <- predict(m5, se.fit=TRUE, test[,-1], type="response")
predicted <- yhat %>% .$fit %>% round()
observed <-test$dtest
cm <- table(predicted, observed) %>% print()
```

```
##          observed
## predicted  0  1
##         0 30  0
##         1  0  5
```

```
MCR <- sum(cm[1, 2], cm[2, 1]) / sum(cm)
print(paste("Misclassification rate:", MCR %>% round(4)))
```

```
## [1] "Misclassification rate: 0"
```

When selecting only the two variables manually, I get constant estiamted degrees of freedom again, with a misclassification rate of 0, again.