Exercise 9 Advanced Methods for Regression and Classification

12433732 - Stefan Merdian

2024-12-15

```r
library(ROCit)
```

```
## Warning: Paket 'ROCit' wurde unter R Version 4.4.2 erstellt
```

```r
data("Diabetes", package="ROCit")
head(Diabetes)
```

```
##     id chol stab.glu hdl ratio glyhb   location age gender height weight  frame
## 1 1000  203       82  56   3.6  4.31 Buckingham  46 female     62    121 medium
## 2 1001  165       97  24   6.9  4.44 Buckingham  29 female     64    218  large
## 3 1002  228       92  37   6.2  4.64 Buckingham  58 female     61    256  large
## 4 1003   78       93  12   6.5  4.63 Buckingham  67   male     67    119  large
## 5 1005  249       90  28   8.9  7.72 Buckingham  64   male     68    183 medium
## 6 1008  248       94  69   3.6  4.81 Buckingham  34   male     71    190  large
##   bp.1s bp.1d bp.2s bp.2d waist hip time.ppn      bmi dtest       whr
## 1   118    59    NA    NA    29  38      720 22.12877     - 0.7631579
## 2   112    68    NA    NA    46  48      360 37.41553     - 0.9583333
## 3   190    92   185    92    49  57      180 48.36549     - 0.8596491
## 4   110    50    NA    NA    33  38      480 18.63600     - 0.8684211
## 5   138    80    NA    NA    44  41      300 27.82202     + 1.0731707
## 6   132    86    NA    NA    36  42      195 26.49673     - 0.8571429
```

```r
dim(Diabetes)
```

```
## [1] 403  22
```

```r
numeric_data <- Diabetes[, sapply(Diabetes, is.numeric)]
na_counts <- colSums(is.na(Diabetes))
print(na_counts)
```

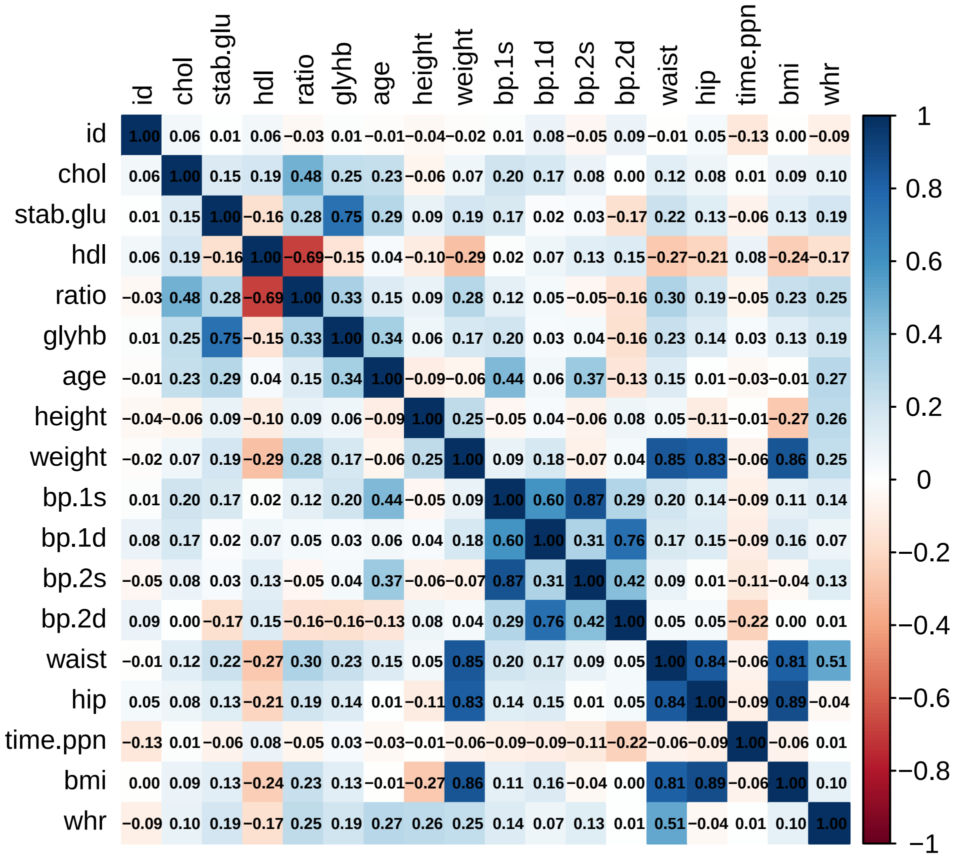
```
##       id     chol stab.glu      hdl    ratio    glyhb location      age
##        0        1        0        1        1       13        0        0
##   gender   height   weight    frame    bp.1s    bp.1d    bp.2s    bp.2d
##        0        5        1       12        5        5      262      262
##    waist      hip time.ppn      bmi    dtest      whr
##        2        2        3        6       13        2
```

```r
library(corrplot)
```

```
## Warning: Paket 'corrplot' wurde unter R Version 4.4.2 erstellt
```

```
## corrplot 0.95 loaded
```

```r
numeric_data <- Diabetes[, sapply(Diabetes, is.numeric)]

cor_matrix <- cor(numeric_data, use = "pairwise.complete.obs")

corrplot(cor_matrix, method = "color",
         tl.col = "black", tl.cex = 0.8, addCoef.col = "black", number.cex = 0.5)
```



	id	chol	stab.glu	hdl	ratio	glyhb	age	height	weight	bp.1s	bp.1d	bp.2s	bp.2d	waist	hip	time.ppn	bmi	whr
id	1.00	0.06	0.01	0.06	−0.03	0.01	−0.01	−0.04	−0.02	0.01	0.08	−0.05	0.09	−0.01	0.05	−0.13	0.00	−0.09
chol	0.06	1.00	0.15	0.19	0.48	0.25	0.23	−0.06	0.07	0.20	0.17	0.08	0.00	0.12	0.08	0.01	0.09	0.10
stab.glu	0.01	0.15	1.00	−0.16	0.28	0.75	0.29	0.09	0.19	0.17	0.02	0.03	−0.17	0.22	0.13	−0.06	0.13	0.19
hdl	0.06	0.19	−0.16	1.00	−0.69	−0.15	0.04	−0.10	−0.29	0.02	0.07	0.13	0.15	−0.27	−0.21	0.08	−0.24	−0.17
ratio	−0.03	0.48	0.28	−0.69	1.00	0.33	0.15	0.09	0.28	0.12	0.05	−0.05	−0.16	0.30	0.19	−0.05	0.23	0.25
glyhb	0.01	0.25	0.75	−0.15	0.33	1.00	0.34	0.06	0.17	0.20	0.03	0.04	−0.16	0.23	0.14	0.03	0.13	0.19
age	−0.01	0.23	0.29	0.04	0.15	0.34	1.00	−0.09	−0.06	0.44	0.06	0.37	−0.13	0.15	0.01	−0.03	−0.01	0.27
height	−0.04	−0.06	0.09	−0.10	0.09	0.06	−0.09	1.00	0.25	−0.05	0.04	−0.06	0.08	0.05	−0.11	−0.01	−0.27	0.26
weight	−0.02	0.07	0.19	−0.29	0.28	0.17	−0.06	0.25	1.00	0.09	0.18	−0.07	0.04	0.85	0.83	−0.06	0.86	0.25
bp.1s	0.01	0.20	0.17	0.02	0.12	0.20	0.44	−0.05	0.09	1.00	0.60	0.87	0.29	0.20	0.14	−0.09	0.11	0.14
bp.1d	0.08	0.17	0.02	0.07	0.05	0.03	0.06	0.04	0.18	0.60	1.00	0.31	0.76	0.17	0.15	−0.09	0.16	0.07
bp.2s	−0.05	0.08	0.03	0.13	−0.05	0.04	0.37	−0.06	−0.07	0.87	0.31	1.00	0.42	0.09	0.01	−0.11	−0.04	0.13
bp.2d	0.09	0.00	−0.17	0.15	−0.16	−0.16	−0.13	0.08	0.04	0.29	0.76	0.42	1.00	0.05	0.05	−0.22	0.00	0.01
waist	−0.01	0.12	0.22	−0.27	0.30	0.23	0.15	0.05	0.85	0.20	0.17	0.09	0.05	1.00	0.84	−0.06	0.81	0.51
hip	0.05	0.08	0.13	−0.21	0.19	0.14	0.01	−0.11	0.83	0.14	0.15	0.01	0.05	0.84	1.00	−0.09	0.89	−0.04
time.ppn	−0.13	0.01	−0.06	0.08	−0.05	0.03	−0.03	−0.01	−0.06	−0.09	−0.09	−0.11	−0.22	−0.06	−0.09	1.00	−0.06	0.01
bmi	0.00	0.09	0.13	−0.24	0.23	0.13	−0.01	−0.27	0.86	0.11	0.16	−0.04	0.00	0.81	0.89	−0.06	1.00	0.10
whr	−0.09	0.10	0.19	−0.17	0.25	0.19	0.27	0.26	0.25	0.14	0.07	0.13	0.01	0.51	−0.04	0.01	0.10	1.00

```r
Diabetes$id <- NULL
Diabetes$height <- NULL
Diabetes$weight <- NULL
Diabetes$location <- NULL
Diabetes$bp.2s <- NULL
Diabetes$bp.2d <- NULL
Diabetes$glyhb <- NULL
Diabetes$ratio <- NULL
Diabetes$whr <- NULL
Diabetes$dtest <- ifelse(Diabetes$dtest == "+", 1, 0)
Diabetes <- na.omit(Diabetes)
```

```r
str(Diabetes)
```

```
## 'data.frame':    366 obs. of  13 variables:
##  $ chol    : int  203 165 228 78 249 248 195 177 263 242 ...
##  $ stab.glu: int  82 97 92 93 90 94 92 87 89 82 ...
##  $ hdl     : int  56 24 37 12 28 69 41 49 40 54 ...
##  $ age     : int  46 29 58 67 64 34 30 45 55 60 ...
##  $ gender  : Factor w/ 2 levels "female","male": 1 1 1 2 2 2 2 2 1 1 ...
##  $ frame   : Factor w/ 3 levels "large","medium",..: 2 1 1 1 2 1 2 1 3 2 ...
##  $ bp.1s   : int  118 112 190 110 138 132 161 160 108 130 ...
##  $ bp.1d   : int  59 68 92 50 80 86 112 80 72 90 ...
##  $ waist   : int  29 46 49 33 44 36 46 34 45 39 ...
##  $ hip     : int  38 48 57 38 41 42 49 40 50 45 ...
##  $ time.ppn: int  720 360 180 480 300 195 720 300 240 300 ...
##  $ bmi     : num  22.1 37.4 48.4 18.6 27.8 ...
##  $ dtest   : num  0 0 0 0 1 0 0 0 0 0 ...
##  - attr(*, "na.action")= 'omit' Named int [1:37] 8 14 28 38 44 51 60 64 65 70 ...
##   ..- attr(*, "names")= chr [1:37] "8" "14" "28" "38" ...
```

Which of the remaining variables should be considered in the model? Argue why it could make sense to exclude predictor variables:

We can exclude the following variables:

- id: It is purely an identifier and holds no relevance to the prediction.
- height and weight: Since BMI is already calculated from these variables, retaining them would introduce multicollinearity without adding new information.
- location: It is not relevant to diabetes and does not contribute meaningfully to the prediction.
- bp.2s, bp.2d: we also removed this column because they have to many na values, otherwise we would eliminate to much data
- The variable glyhb is already reflected in dtest, making it redundant. Similarly, ratio is derived as the ratio between chol and hdl, which makes it highly correlated with these variables. The same applies to whr, as it is calculated using waist and hip, leading to strong correlation.

```r
set.seed(123)
n <- nrow(Diabetes)
train_indices <- sample(1:n, size = floor(2 * n / 3))
train_data <- Diabetes[train_indices, ]
test_data <- Diabetes[-train_indices, ]
```

1) Logistic regression model

```r
library(caret)
```

```
## Warning: Paket 'caret' wurde unter R Version 4.4.2 erstellt
```

```
## Lade nötiges Paket: ggplot2
```

```
## Warning: Paket 'ggplot2' wurde unter R Version 4.4.2 erstellt
## Lade nötiges Paket: lattice
```

```
model <- glm(dtest ~ ., data = train_data, family = "binomial")
summary(model)
```

```
## 
## Call:
## glm(formula = dtest ~ ., family = "binomial", data = train_data)
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.102e+01  3.960e+00  -2.782   0.0054 ** 
## chol         6.154e-03  6.662e-03   0.924   0.3556    
## stab.glu     2.884e-02  5.040e-03   5.723 1.05e-08 ***
## hdl         -2.042e-02  1.688e-02  -1.209   0.2265    
## age          1.717e-02  2.022e-02   0.849   0.3958    
## gendermale  -6.930e-01  7.217e-01  -0.960   0.3369    
## framemedium  2.386e-01  6.391e-01   0.373   0.7088    
## framesmall   5.046e-01  8.658e-01   0.583   0.5600    
## bp.1s        1.935e-02  1.669e-02   1.159   0.2463    
## bp.1d       -1.049e-02  2.915e-02  -0.360   0.7191    
## waist        1.244e-01  9.641e-02   1.291   0.1968    
## hip         -6.386e-02  1.198e-01  -0.533   0.5941    
## time.ppn     1.623e-03  8.139e-04   1.994   0.0462 *  
## bmi          1.814e-03  9.048e-02   0.020   0.9840    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 214.43  on 243  degrees of freedom
## Residual deviance: 110.13  on 230  degrees of freedom
## AIC: 138.13
## 
## Number of Fisher Scoring iterations: 6
```

```
predicted_probs <- predict(model, newdata = test_data, type = "response")
predicted_class <- ifelse(predicted_probs > 0.5, 1, 0)
conf_matrix <- table(Predicted = predicted_class, Actual = test_data$dtest)
misclassification_rate <- sum(predicted_class != test_data$dtest) / nrow(test_data)


print(conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 104   7
##         1   1  10
```

```
cat("Misclassification Rate:", misclassification_rate, "\n")
```

```
## Misclassification Rate: 0.06557377
```

Which problems do you face?:

- Multicollinearity: Many variables exhibited high correlation (e.g., ratio, whr), leading to singularity issues. I used the correlation matrix to identify and address this.
- Binary Conversion: The target variable had + and - signs, which needed to be converted into a binary format (0/1).

2) Sparse logistic regression model

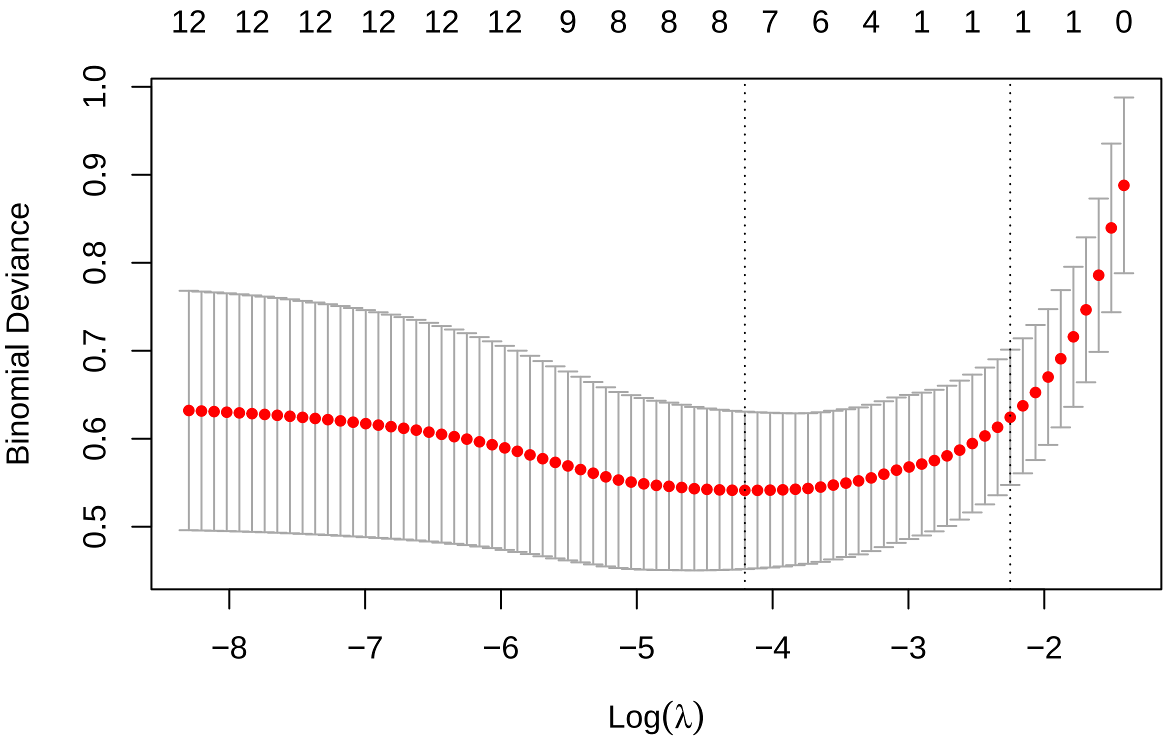
```
library(glmnet)
```

```
## Warning: Paket 'glmnet' wurde unter R Version 4.4.2 erstellt
```

```
## Lade nötiges Paket: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
cv_model <- cv.glmnet(model.matrix(dtest ~ ., data = train_data) , train_data$dtest, family = "binomial
plot(cv_model)
```



```r
predicted_probs <- predict(cv_model, newx = model.matrix(dtest ~ ., data = train_data), s = "lambda.min
predicted_class <- ifelse(predicted_probs > 0.5, 1, 0)

conf_matrix <- table(Predicted = predicted_class, Actual = train_data$dtest)
print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```r
print(conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 201  17
##         1   4  22
```

```r
misclassification_rate <- sum(predicted_class != train_data$dtest) / length(train_data$dtest)
cat("Misclassification Rate:", misclassification_rate, "\n")
```

```
## Misclassification Rate: 0.08606557
```

```r
predicted_probs <- predict(cv_model, newx = model.matrix(dtest ~ ., data = test_data), s = "lambda.min"
predicted_class <- ifelse(predicted_probs > 0.5, 1, 0)

conf_matrix <- table(Predicted = predicted_class, Actual = test_data$dtest)
print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```r
print(conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 103   8
##         1   2   9
```

```r
misclassification_rate <- sum(predicted_class != test_data$dtest) / length(test_data$dtest)
cat("Misclassification Rate:", misclassification_rate, "\n")
```

```
## Misclassification Rate: 0.08196721
```

The results of the sparse logistic regression using cv.glmnet on the train and test datasets show perfect performance.

For the training data:

- 201 true negatives and 17 false negative
- 22 true positives and 4 false postive
- MKR: 8.66%

For the test data, the confusion matrix shows:

- 103 true negatives and 8 false negative
- 9 true positives and 2 false postive
- MKR: 8.2%

3) GAM models

a)

```
library(mgcv)
```

```
## Lade nötiges Paket: nlme

## This is mgcv 1.9-1. For overview type ’help("mgcv-package")’.
```

```
gam_model1 <- gam(dtest ~ s(age) + s(bmi) + s(stab.glu) + s(waist) + s(bp.1s) + s(bp.1d) +
                        s(chol) + s(hdl)  + s(time.ppn) +
                        gender + frame,
                      family = "binomial",
                      data = train_data)
summary(gam_model1)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(age) + s(bmi) + s(stab.glu) + s(waist) + s(bp.1s) +
##     s(bp.1d) + s(chol) + s(hdl) + s(time.ppn) + gender + frame
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.6673     3.4495  -1.933   0.0533 .
## gendermale   -0.3352     1.0451  -0.321   0.7484
## framemedium   1.3731     1.0020   1.370   0.1706
## framesmall    1.5385     1.2574   1.223   0.2211
## ---
## 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(age)      1.679  2.097  1.291   0.526
## s(bmi)      1.000  1.000  0.738   0.390
## s(stab.glu) 5.082  6.114 28.429 8.4e-05 ***
## s(waist)    1.000  1.000  2.180   0.140
## s(bp.1s)    1.533  1.883  1.819   0.318
## s(bp.1d)    1.242  1.438  1.935   0.335
## s(chol)     2.077  2.679  3.637   0.243
## s(hdl)      5.969  6.410  5.670   0.593
## s(time.ppn) 3.857  4.699  7.513   0.168
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.778   Deviance explained = 74.9%
## UBRE = -0.55475  Scale est. = 1         n = 244
```

```r
predicted_probs <- predict(gam_model1, newdata = train_data, type = "response")
predicted_class <- ifelse(predicted_probs > 0.5, 1, 0)

conf_matrix <- table(Predicted = predicted_class, Actual = train_data$dtest)
print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```r
print(conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 203   7
##         1   2  32
```

```r
misclassification_rate <- sum(predicted_class != train_data$dtest) / nrow(train_data)
cat("Misclassification Rate:", misclassification_rate, "\n")
```

```
## Misclassification Rate: 0.03688525
```

b)

```r
gam_model2 <- gam(dtest ~ s(age, k = 5) + s(bmi, k = 5) + s(stab.glu, k = 5) +
                         s(waist, k = 5)  + s(bp.1s, k = 5) +
                         s(bp.1d, k = 5) +
                         s(chol, k = 5) + s(hdl, k = 5) +
                          s(time.ppn, k = 5) +
                         gender + frame,
                       family = binomial,
                       data = train_data)
summary(gam_model2)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(age, k = 5) + s(bmi, k = 5) + s(stab.glu, k = 5) +
##     s(waist, k = 5) + s(bp.1s, k = 5) + s(bp.1d, k = 5) + s(chol,
##     k = 5) + s(hdl, k = 5) + s(time.ppn, k = 5) + gender + frame
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.4234     1.0410  -4.249 2.14e-05 ***
## gendermale    0.1131     0.8106   0.140    0.889
## framemedium   1.1301     0.8141   1.388    0.165
## framesmall    1.4864     1.0694   1.390    0.165
## ---
## 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(age)       1.000  1.000  0.542  0.4616
## s(bmi)       1.000  1.000  0.162  0.6873
## s(stab.glu) 2.754  3.231 30.329 2.6e-06 ***
## s(waist)     1.000  1.000  0.800  0.3711
## s(bp.1s)     1.121  1.228  1.092  0.3147
## s(bp.1d)     1.000  1.000  0.188  0.6645
## s(chol)      2.681  3.236  6.817  0.0853 .
## s(hdl)       2.071  2.566  2.859  0.2964
## s(time.ppn) 3.759  3.960 10.271  0.0341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.688   Deviance explained = 65.3%
## UBRE = -0.52787  Scale est. = 1         n = 244
```

```
predicted_probs <- predict(gam_model2, newdata = train_data, type = "response")
predicted_class <- ifelse(predicted_probs > 0.5, 1, 0)

conf_matrix <- table(Predicted = predicted_class, Actual = train_data$dtest)
print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```
print(conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 202   9
##         1   3  30
```

```
misclassification_rate <- sum(predicted_class != train_data$dtest) / nrow(train_data)
cat("Misclassification Rate:", misclassification_rate, "\n")
```

```
## Misclassification Rate: 0.04918033
```

c)

```
summary(gam_model1)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(age) + s(bmi) + s(stab.glu) + s(waist) + s(bp.1s) +
##     s(bp.1d) + s(chol) + s(hdl) + s(time.ppn) + gender + frame
```

```
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.6673     3.4495  -1.933   0.0533 .
## gendermale   -0.3352     1.0451  -0.321   0.7484
## framemedium   1.3731     1.0020   1.370   0.1706
## framesmall    1.5385     1.2574   1.223   0.2211
## ---
## 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(age)      1.679  2.097  1.291   0.526
## s(bmi)      1.000  1.000  0.738   0.390
## s(stab.glu) 5.082  6.114 28.429 8.4e-05 ***
## s(waist)    1.000  1.000  2.180   0.140
## s(bp.1s)    1.533  1.883  1.819   0.318
## s(bp.1d)    1.242  1.438  1.935   0.335
## s(chol)     2.077  2.679  3.637   0.243
## s(hdl)      5.969  6.410  5.670   0.593
## s(time.ppn) 3.857  4.699  7.513   0.168
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.778   Deviance explained = 74.9%
## UBRE = -0.55475  Scale est. = 1         n = 244
```

In this GAM, the variable s(stab.glu) remains the only statistically significant smooth term, with a p-value < 0.01. Its effective degrees of freedom (edf) is 5.082, indicating a more flexible and moderately non-linear relationship with the response variable.

The smooth term s(hdl) appears not significant with a p-value of 0.59, but its edf is 5.96, showing slight non-linearity.

The remaining smooth terms, such as s(age), s(bmi), s(waist), s(bp.1d), are not statistically significant, with p-values well above 0.05. Their edf values are close to 1, implying these relationships are effectively linear or have no meaningful effect on the response.

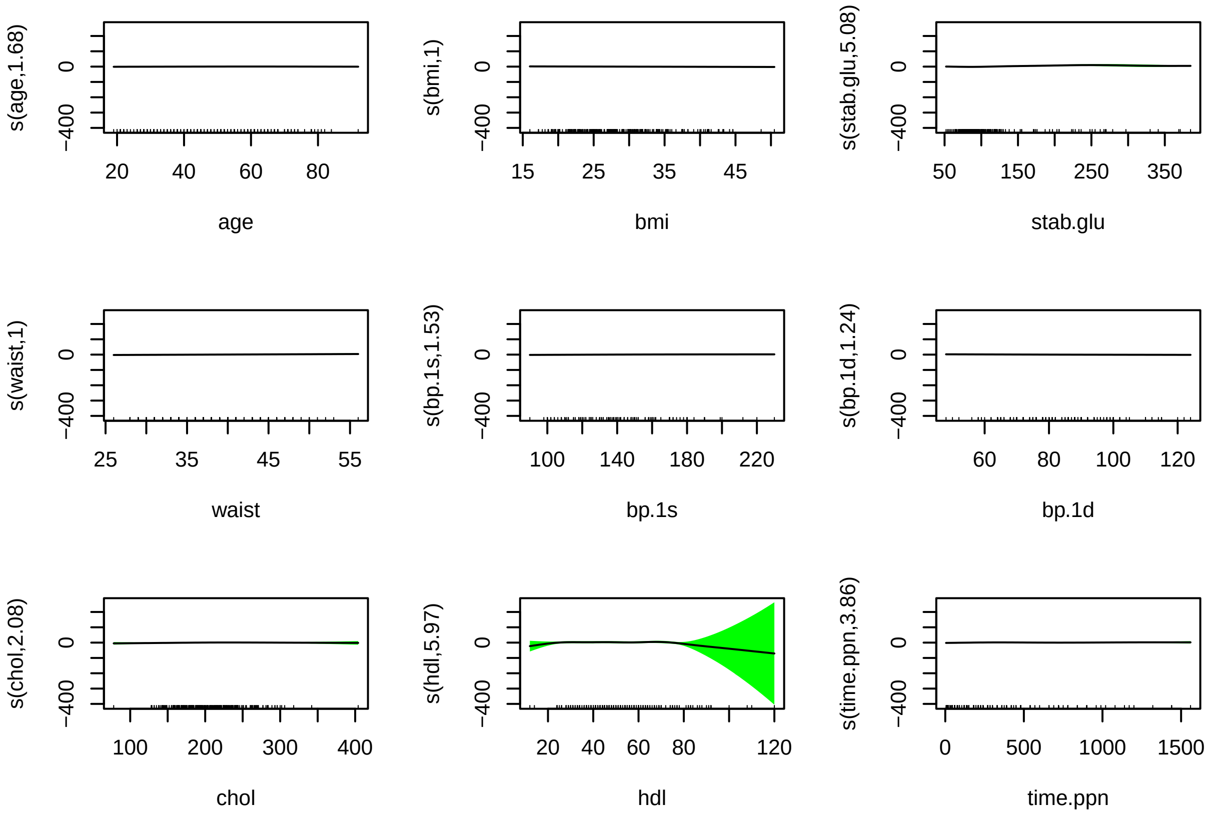
```r
summary(gam_model2)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(age, k = 5) + s(bmi, k = 5) + s(stab.glu, k = 5) +
##     s(waist, k = 5) + s(bp.1s, k = 5) + s(bp.1d, k = 5) + s(chol,
##     k = 5) + s(hdl, k = 5) + s(time.ppn, k = 5) + gender + frame
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.4234     1.0410  -4.249 2.14e-05 ***
## gendermale    0.1131     0.8106   0.140    0.889
## framemedium   1.1301     0.8141   1.388    0.165
## framesmall    1.4864     1.0694   1.390    0.165
## ---
## 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(age)      1.000  1.000  0.542  0.4616
## s(bmi)      1.000  1.000  0.162  0.6873
## s(stab.glu) 2.754  3.231 30.329 2.6e-06 ***
## s(waist)    1.000  1.000  0.800  0.3711
## s(bp.1s)    1.121  1.228  1.092  0.3147
## s(bp.1d)    1.000  1.000  0.188  0.6645
## s(chol)     2.681  3.236  6.817  0.0853 .
## s(hdl)      2.071  2.566  2.859  0.2964
## s(time.ppn) 3.759  3.960 10.271  0.0341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.688   Deviance explained = 65.3%
## UBRE = -0.52787  Scale est. = 1         n = 244
```

For the second model, the results are quite similar to the previous one, but two variables, s(time.ppn) and s(chol), have become more significant. Additionally, the effective degrees of freedom (edf) appear more normalized, likely due to the restriction on degrees of freedom (k = 5). However, the overall deviance explained is slightly lower compared to the previous model.

d)

```
par(mar = c(5, 4, 2, 2))
plot(gam_model1, page = 1, shade = TRUE, shade.col = "green")
```



All terms are kinda flat or linear. Just hdl and stab.glu has some non-linear relation to dependent variable. The green shaded areas (confidence intervals) are wide for certain predictors, particularly at the edges of the range. This suggests high uncertainty due to a lack of sufficient data in those regions (hdl). The model struggles to identify non-linear patterns.

e)

```
predicted_probs1 <- predict(gam_model1, newdata = test_data, type = "response")
predicted_class1 <- ifelse(predicted_probs1 > 0.5, 1, 0)


conf_matrix1 <- table(Predicted = predicted_class1, Actual = test_data$dtest)
print("Confusion Matrix1:")
```

```
## [1] "Confusion Matrix1:"
```

```
print(conf_matrix1)
```

```
##          Actual
## Predicted  0  1
##         0 98  8
##         1  7  9
```

```
misclassification_rate1 <- sum(predicted_class1 != test_data$dtest) / nrow(test_data)
cat("Misclassification Rate1:", misclassification_rate1, "\n")
```

```
## Misclassification Rate1: 0.1229508
```

First Confusion Matrix (GAM without degree restriction): - The model correctly predicted 98 instances as class 0 and 9 instances as class 1. - There were 8 false positives and 7 false negatives . - The misclassification rate is 12.3%.

```
predicted_probs2<- predict(gam_model2, newdata = test_data, type = "response")
predicted_class2<- ifelse(predicted_probs2 > 0.5, 1, 0)

conf_matrix2 <- table(Predicted = predicted_class2, Actual = test_data$dtest)
print("Confusion Matrix2:")
```

```
## [1] "Confusion Matrix2:"
```

```
print(conf_matrix2)
```

```
##          Actual
## Predicted   0   1
##         0 101   6
##         1   4  11
```

```
misclassification_rate2 <- sum(predicted_class2 != test_data$dtest) / nrow(test_data)
cat("Misclassification Rate2:", misclassification_rate2, "\n")
```

```
## Misclassification Rate2: 0.08196721
```

Second Confusion Matrix (GAM with degree restriction k = 5) same here:

- The model correctly predicted 99 instances as class 0 and 12 instances as class 1.
- There were 6 false positives and 5 false negatives .
- The misclassification rate is 9.01%.

The second model appears to be better, likely because it is more generalized and avoids overfitting to the training data.

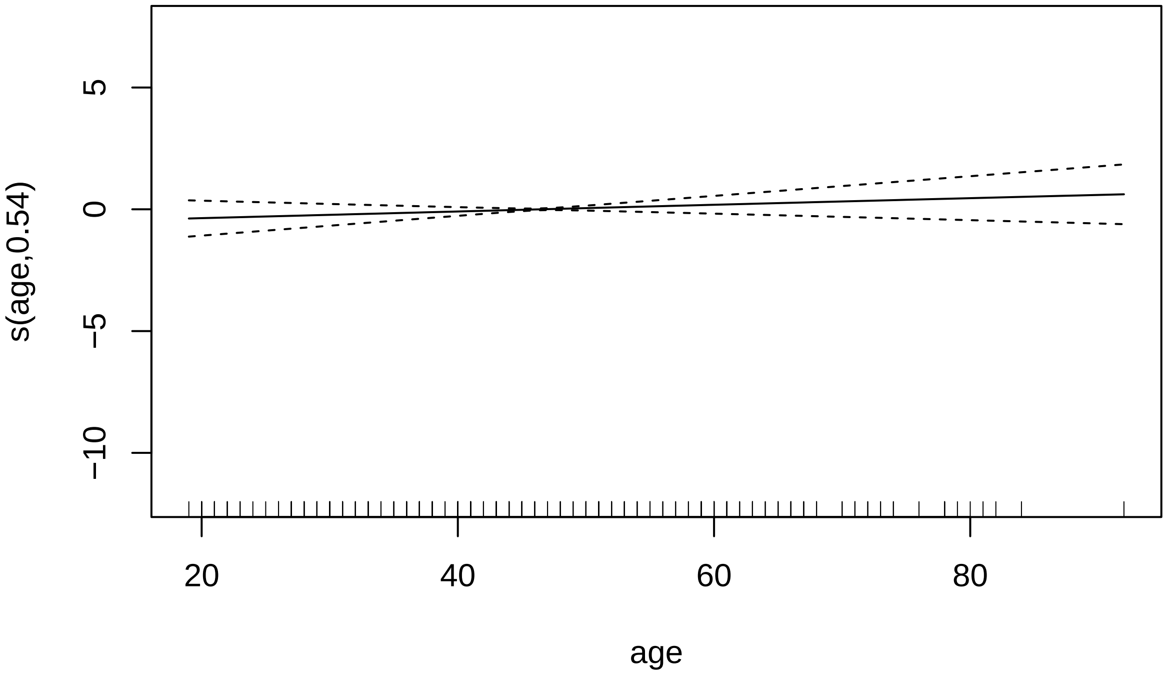
f)

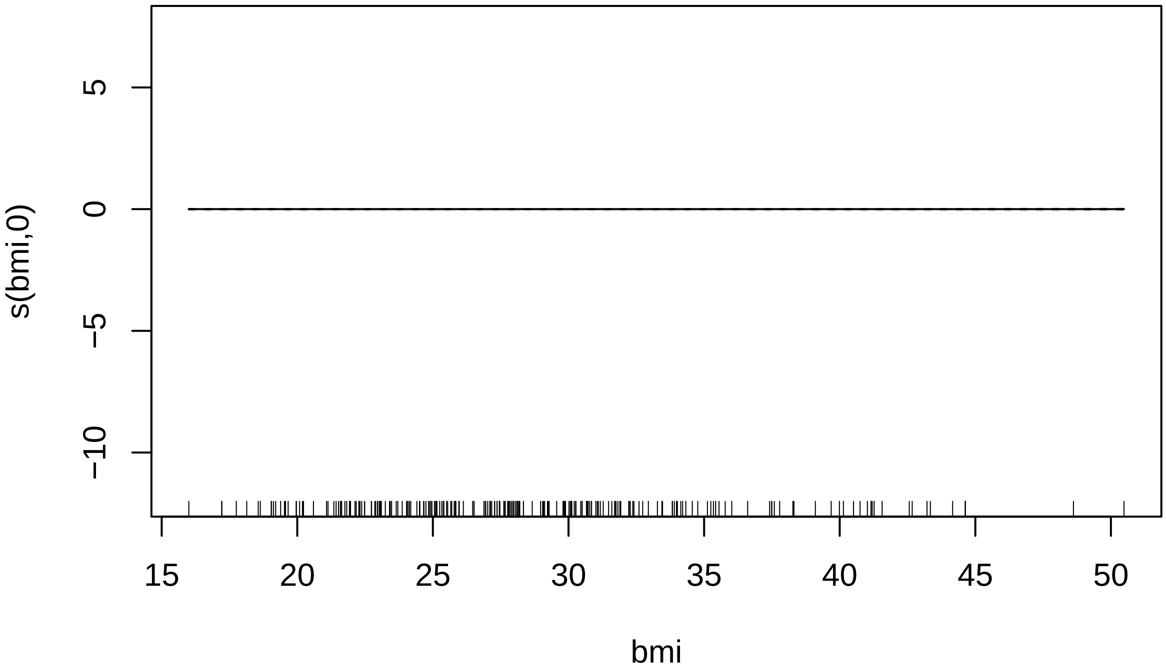
We will use thin-plate regression splines with shrinkage (bs = “ts”). The select = TRUE option applies automatic smoothing parameter selection, allowing insignificant smooth terms to shrink effectively toward zero. The model uses the REML method for robust estimation of smoothness penalties.

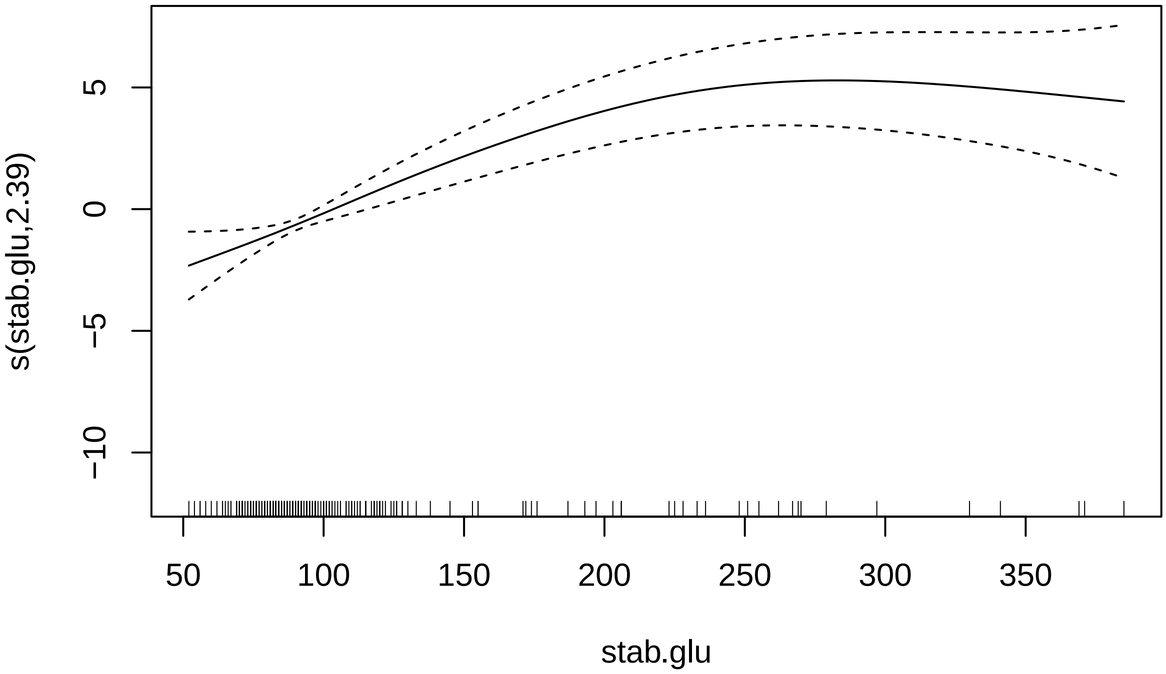
```
gam_model_select <- gam(dtest ~ s(age, bs = "ts") + s(bmi, bs = "ts") +
                        s(stab.glu, bs = "ts") + s(waist, bs = "ts") +
                        s(bp.1s, bs = "ts") + s(bp.1d, bs = "ts") +
                        s(chol, bs = "ts") + s(hdl, bs = "ts") +
                        s(time.ppn, bs = "ts") + gender + frame,
                        family = binomial, data = train_data,
                        method = "REML", select = TRUE)

summary(gam_model_select)
```

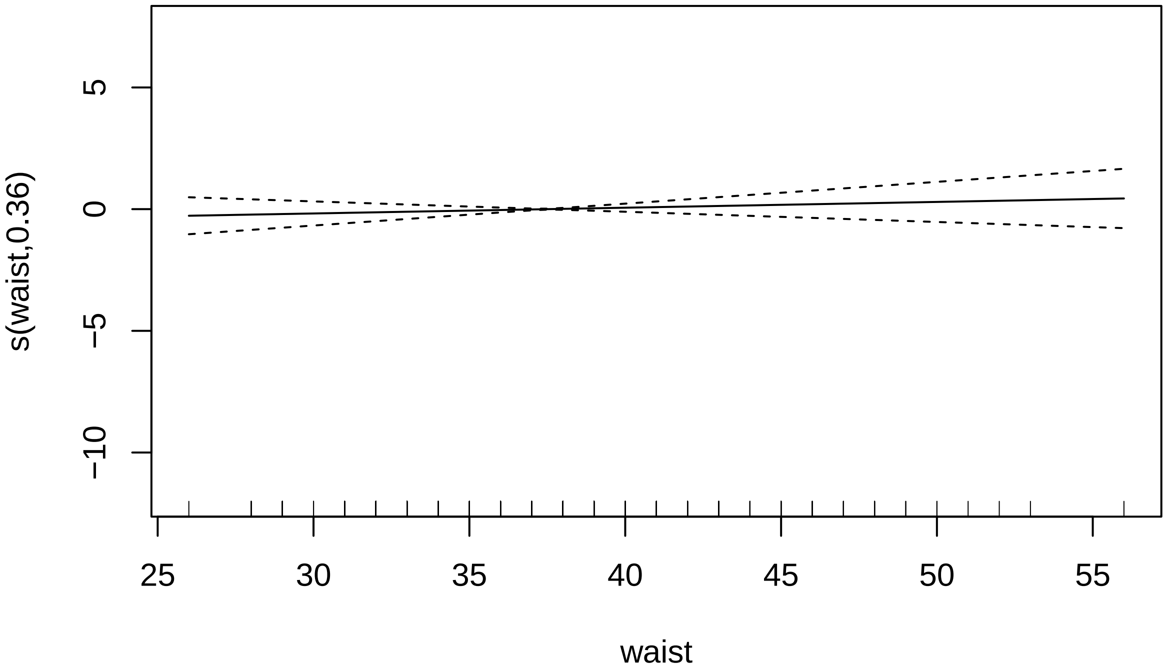
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## dtest ~ s(age, bs = "ts") + s(bmi, bs = "ts") + s(stab.glu, bs = "ts") +
##     s(waist, bs = "ts") + s(bp.1s, bs = "ts") + s(bp.1d, bs = "ts") +
##     s(chol, bs = "ts") + s(hdl, bs = "ts") + s(time.ppn, bs = "ts") +
##     gender + frame
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6840     0.6477  -4.144 3.42e-05 ***
## gendermale   -0.4294     0.6134  -0.700    0.484
## framemedium   0.2853     0.6328   0.451    0.652
## framesmall    0.1868     0.8392   0.223    0.824
## ---
## 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(age)      5.412e-01      9  1.038  0.1541
## s(bmi)      1.664e-05      9  0.000  0.7778
## s(stab.glu) 2.388e+00      9 42.671  <2e-16 ***
## s(waist)    3.553e-01      9  0.520  0.2209
## s(bp.1s)    6.590e-01      9  1.770  0.0924 .
## s(bp.1d)    1.902e-05      9  0.000  0.9555
## s(chol)     4.952e-01      9  0.914  0.1615
## s(hdl)      2.419e+00      9  6.141  0.0424 *
## s(time.ppn) 6.250e-01      9  1.472  0.1216
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =    0.6   Deviance explained = 55.6%
## -REML = 62.131  Scale est. = 1         n = 244
```

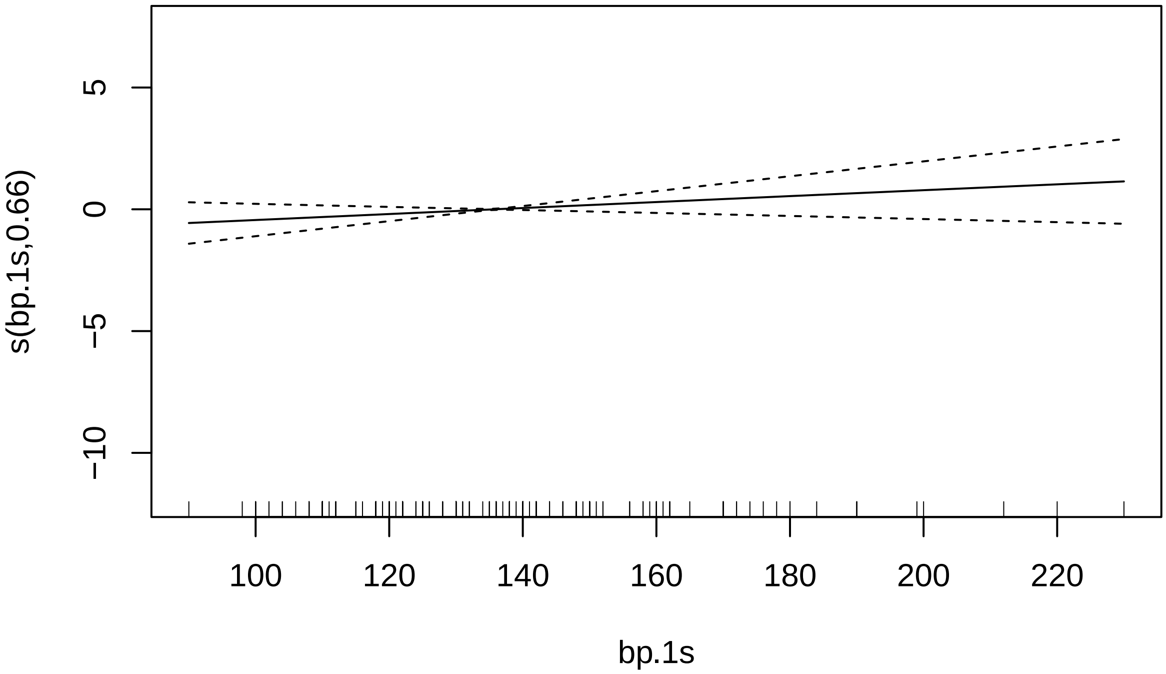
```
plot(gam_model_select)
```

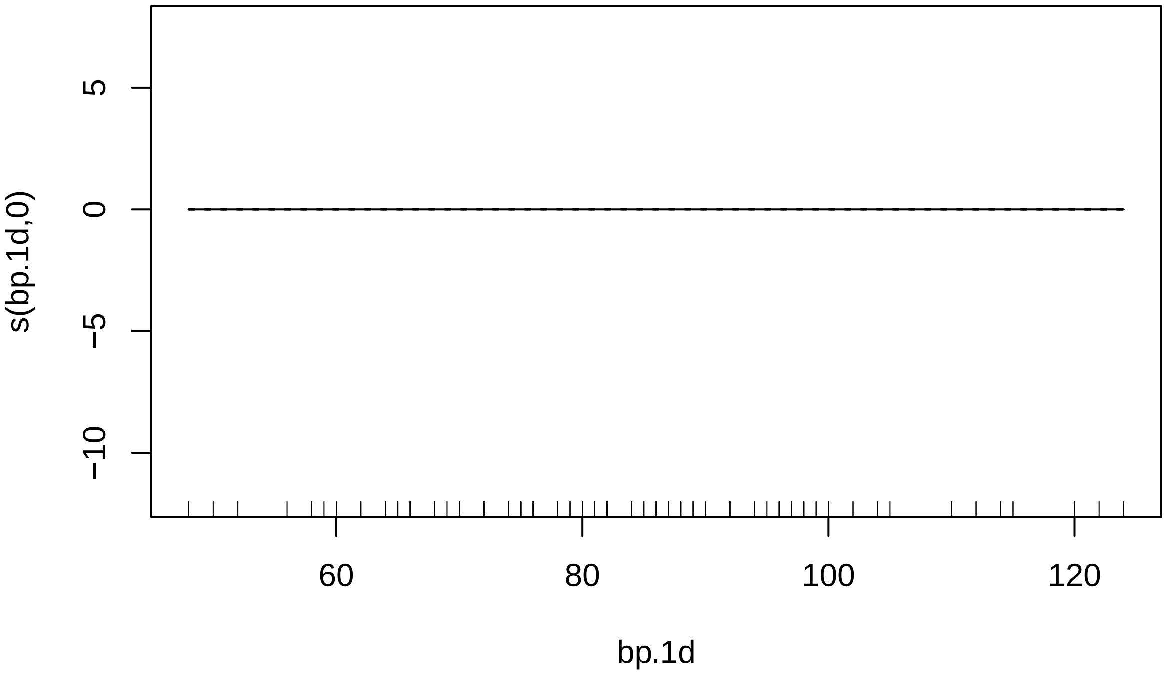

s(age,0.54)
5
0
−5
−10
20
40
60
80
age


s(bmi,0)
5
0
−5
−10
15
20
25
30
35
40
45
50
bmi

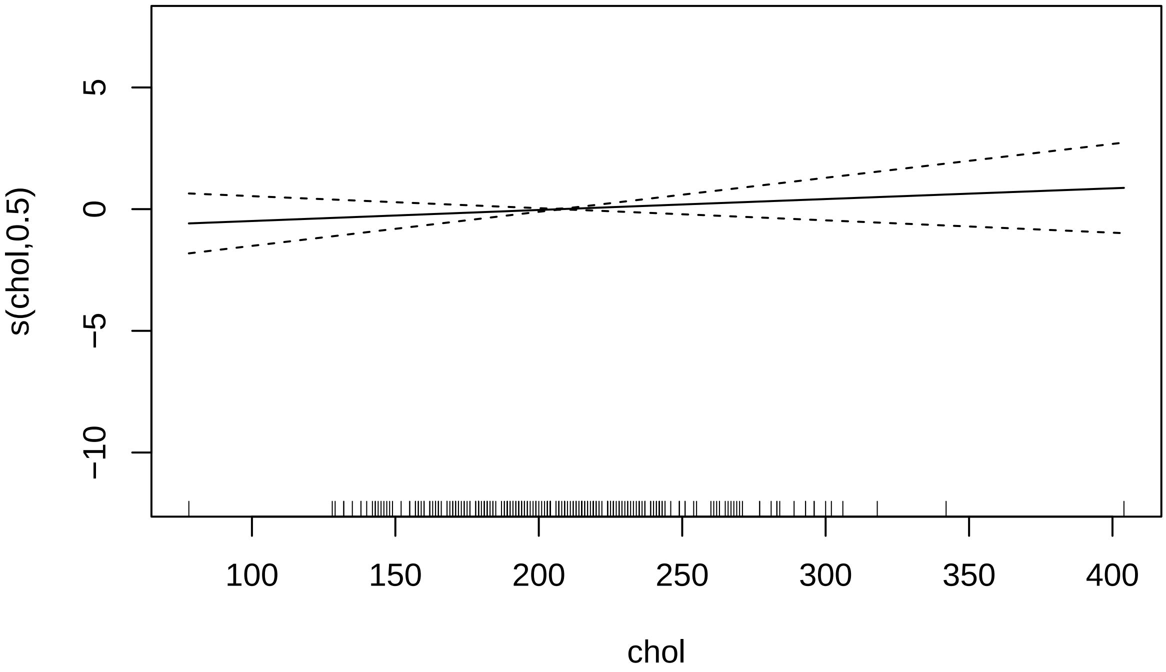

s(stab.glu,2.39)
5
0
−5
−10
50
100
150
200
250
300
350
stab.glu

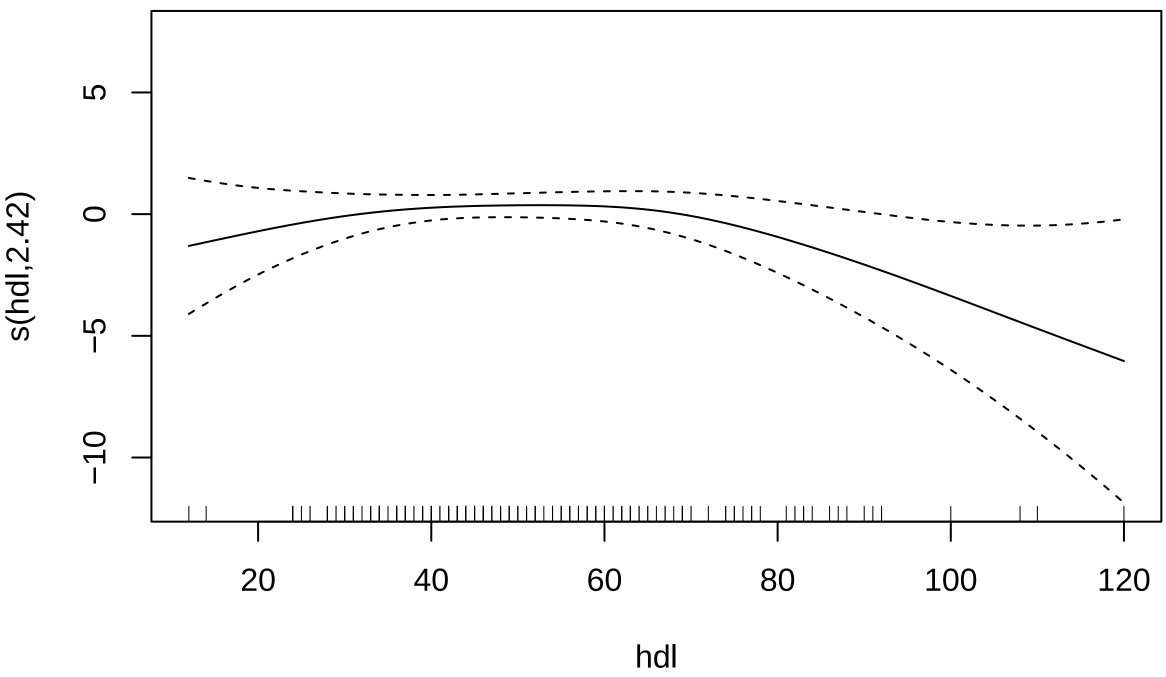

5
0
−5
−10
s(waist,0.36)
25
30
35
40
45
50
55
waist

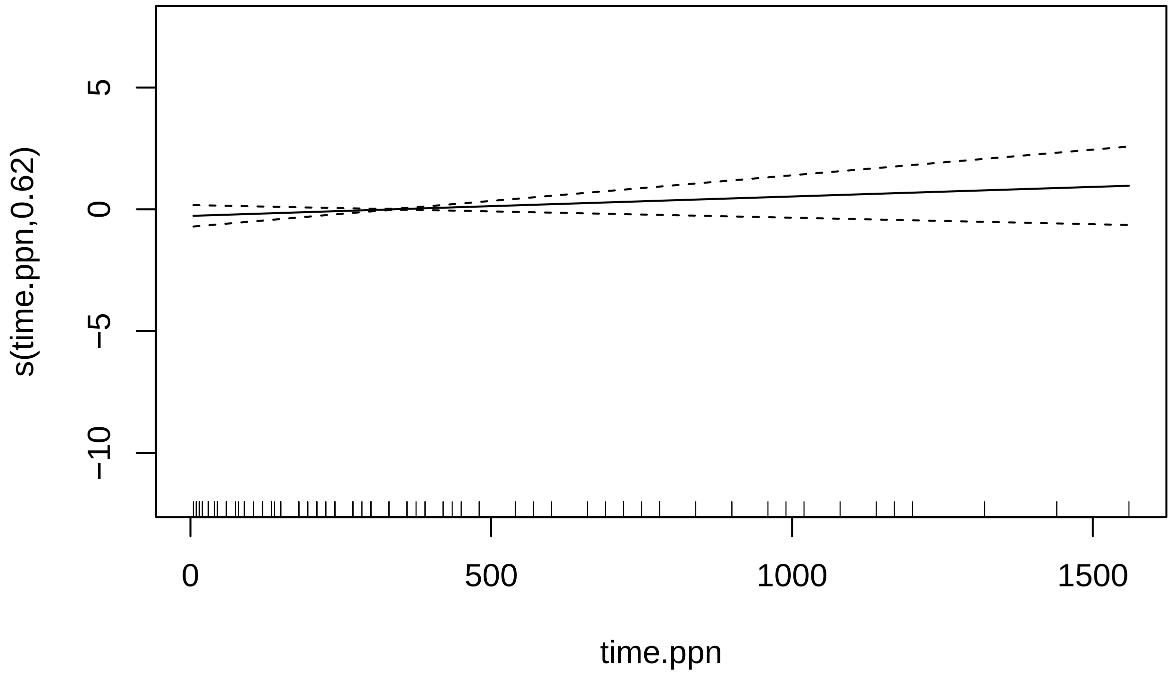

5
0
−5
−10
s(bp.1s,0.66)
100
120
140
160
180
200
220
bp.1s



s(bp.1d,0)
5
0
−5
−10
60
80
100
120
bp.1d


s(chol,0.5)
5
0
−5
−10
100
150
200
250
300
350
400
chol


5
0
−5
−10
s(hdl,2.42)
20
40
60
80
100
120
hdl



Based on the summary and plots, stab.glu remains the most significant variable. Additionally, hdl and bp.1s show slight improvements in significance. Therefore, we will select these variables for the shrunk model.

```
predicted_probs <- predict(gam_model_select, newdata = test_data, type = "response")
predicted_class <- ifelse(predicted_probs > 0.5, 1, 0)

# Confusion Matrix and Misclassification Rate
conf_matrix <- table(Predicted = predicted_class, Actual = test_data$dtest)
print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```
print(conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 103   7
##         1   2  10
```

```
misclassification_rate <- sum(predicted_class != test_data$dtest) / nrow(test_data)
cat("Misclassification Rate:", misclassification_rate, "\n")
```

```
## Misclassification Rate: 0.07377049
```

This is currently the best model so far, with a misclassification rate of 7.4%. In comparison, the other models had:

- 12.3% for the standard GAM model,
- 9.01% for the model with restricted degrees of freedom.

This shows that the shrinkage approach effectively improved the performance by enhancing the impact of significant variables while reducing the influence of less important ones.

g)

```
shrinked_model <- gam(dtest ~s(stab.glu)  + s(bp.1s) +
                         s(hdl),
                      family = "binomial",
                      data = train_data)

predicted_probs <- predict(shrinked_model, newdata = test_data, type = "response")
predicted_class <- ifelse(predicted_probs > 0.5, 1, 0)

# Confusion Matrix and Misclassification Rate
conf_matrix <- table(Predicted = predicted_class, Actual = test_data$dtest)
print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```
print(conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 102   9
##         1   3   8
```

```
misclassification_rate <- sum(predicted_class != test_data$dtest) / nrow(test_data)
cat("Misclassification Rate:", misclassification_rate, "\n")
```

```
## Misclassification Rate: 0.09836066
```

After selecting only the three most significant variables from the previous exercise, our predictions on the test set are quite similar to the earlier model, which included many more variables. Although the performance is slightly worse, we have greatly simplified the model while achieving nearly the same results as before.