GAM -Example on Car data

This study completes the previous one in which we explored the of a GLM regression. Now, we will use a Generalized Additive Model, aka GAM, and see the benefits of this model. We use the same data from last time.

```
# Define columnn class for dataset
colCls <- c("integer",         # row id
            "character",       # analysis year
            "numeric",         # exposure
            "character",       # new business / renewal business
            "numeric",         # driver age (continuous)
            "character",       # driver age (categorical)
            "character",       # driver gender
            "character",       # marital status
            "numeric",         # years licensed (continuous)
            "character",       # years licensed (categorical)
            "character",       # ncd level
            "character",       # region
            "character",       # body code
            "numeric",         # vehicle age (continuous)
            "character",       # vehicle age (categorical)
            "numeric",         # vehicle value
            "character",       # seats
            rep("numeric", 6), # ccm, hp, weight, length, width, height (all continuous)
            "character",       # fuel type
            rep("numeric", 3)  # prior claims, claim count, claim incurred (all continuous)
)
```

```
# Define the data path and filename
data.path <- "C:\\Users\\William.Tiritilli\\Documents\\Project P\\Frees\\Tome 2 - Chapter 1\\"
data.fn <- "sim-modeling-dataset2.csv"

# Read in the data with the appropriate column classes
dta <- read.csv(paste(data.path, data.fn, sep = "/"),
                colClasses = colCls)
str(dta)
```

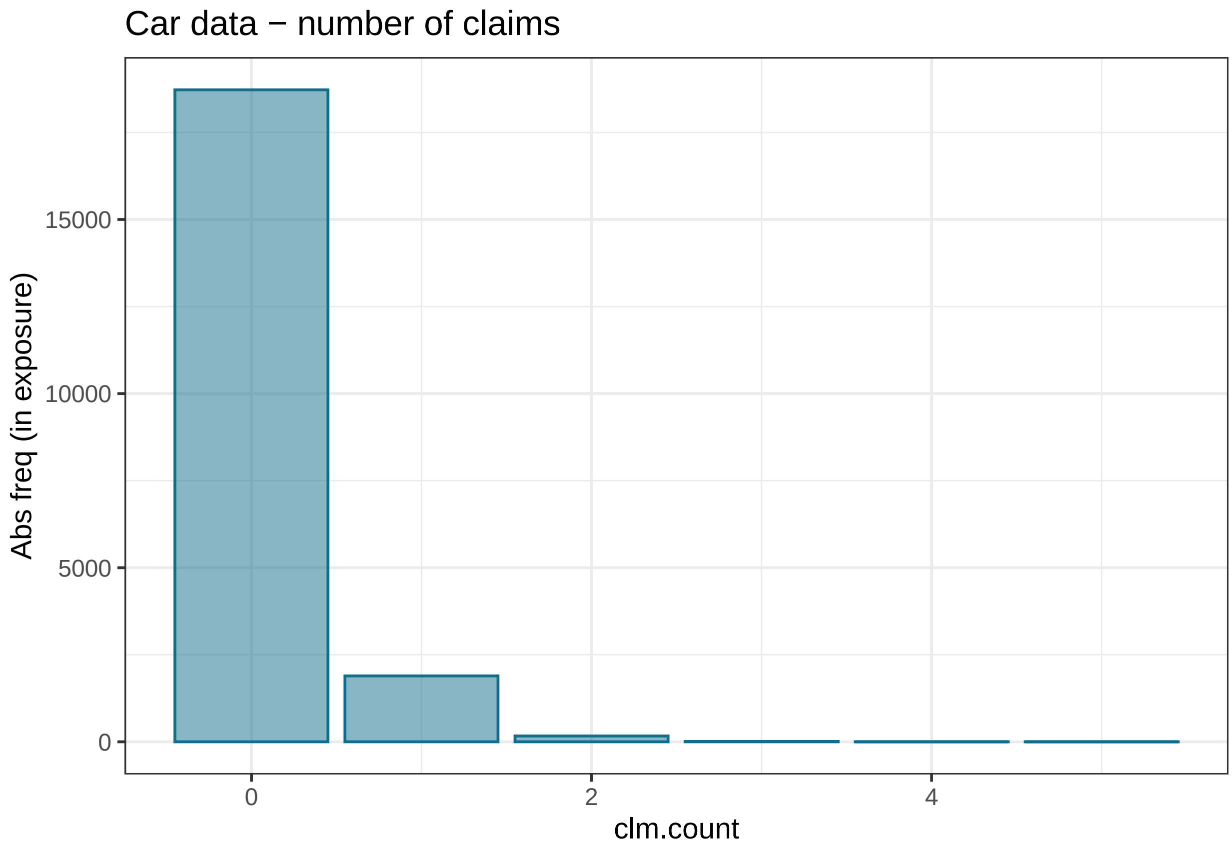
```
## 'data.frame':    40760 obs. of  27 variables:
##  $ row.id        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ year          : chr  "2010" "2010" "2010" "2010" ...
##  $ exposure      : num  1 1 1 0.08 1 0.08 1 1 0.08 1 ...
##  $ nb.rb         : chr  "RB" "NB" "RB" "RB" ...
##  $ driver.age    : num  63 33 68 68 68 68 53 68 68 65 ...
##  $ drv.age       : chr  "63" "33" "68" "68" ...
##  $ driver.gender : chr  "Male" "Male" "Male" "Male" ...
##  $ marital.status: chr  "Married" "Married" "Married" "Married" ...
```

```
##  $ yrs.licensed  : num  5 1 2 2 2 2 5 2 2 2 ...
##  $ yrs.lic       : chr  "5" "1" "2" "2" ...
##  $ ncd.level     : chr  "6" "5" "4" "4" ...
##  $ region        : chr  "3" "38" "33" "33" ...
##  $ body.code     : chr  "A" "B" "C" "C" ...
##  $ vehicle.age   : num  3 3 2 2 1 1 3 1 1 5 ...
##  $ veh.age       : chr  "3" "3" "2" "2" ...
##  $ vehicle.value : num  21.4 17.1 17.3 17.3 25 ...
##  $ seats         : chr  "5" "3" "5" "5" ...
##  $ ccm           : num  1248 2476 1948 1948 1461 ...
##  $ hp            : num  70 94 90 90 85 85 70 85 85 65 ...
##  $ weight        : num  1285 1670 1760 1760 1130 ...
##  $ length        : num  4.32 4.79 4.91 4.91 4.04 ...
##  $ width         : num  1.68 1.74 1.81 1.81 1.67 ...
##  $ height        : num  1.8 1.97 1.75 1.75 1.82 ...
##  $ fuel.type     : chr  "Diesel" "Diesel" "Diesel" "Diesel" ...
##  $ prior.claims  : num  0 0 0 0 0 0 4 0 0 0 ...
##  $ clm.count     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ clm.incurred  : num  0 0 0 0 0 0 0 0 0 0 ...
```

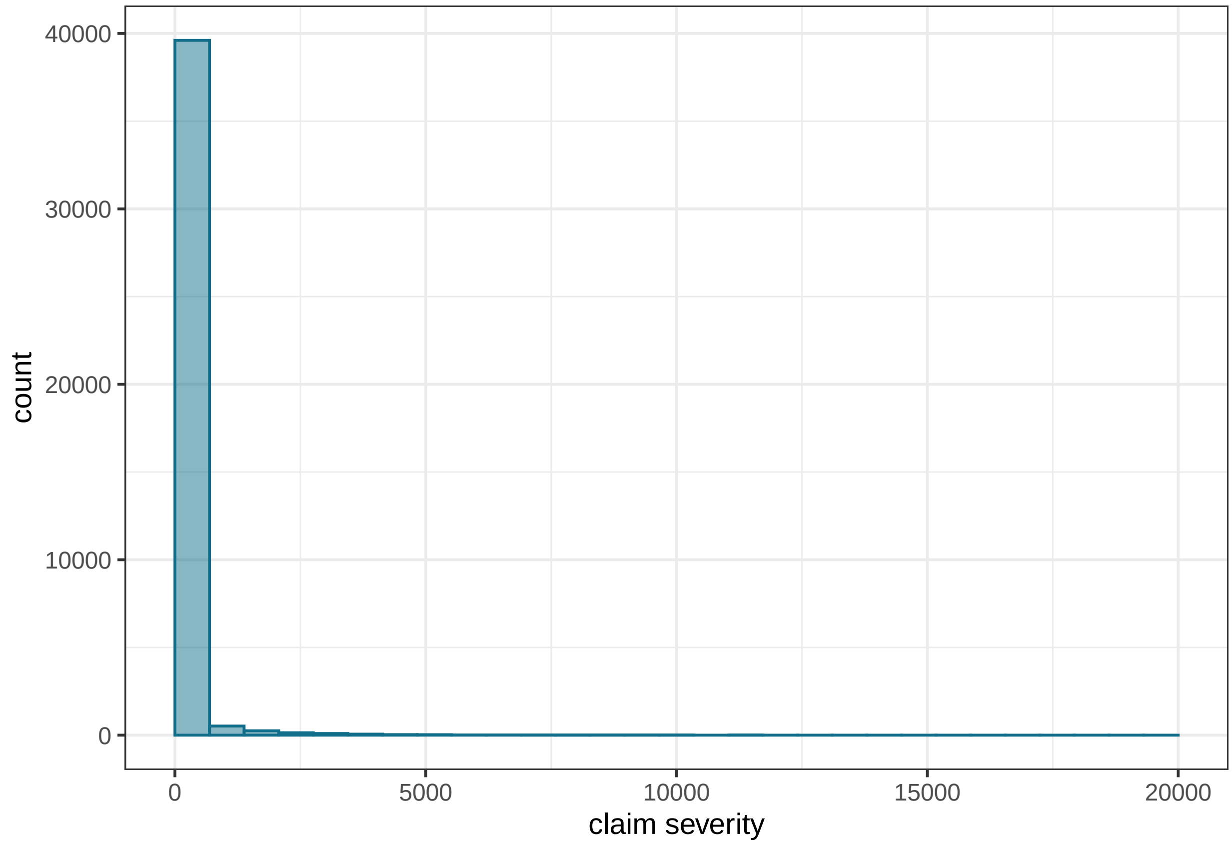
```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```r
KULbg <- "#116E8A"

g_freq <- ggplot(dta, aes(clm.count)) + theme_bw() +
  geom_bar(aes(weight = exposure), col = KULbg,
           fill = KULbg, alpha = .5) +
  labs(y = "Abs freq (in exposure)") +
  ggtitle("Car data - number of claims")
g_freq
```




```r
g_sev <- ggplot(dta, aes(x = clm.incurred)) + theme_bw() +
  geom_histogram(bins = 30, boundary = 0, color = KULbg, fill = KULbg, alpha = .5) +
  labs(x = "claim severity") +
  xlim(c(0, 20000))
g_sev
```



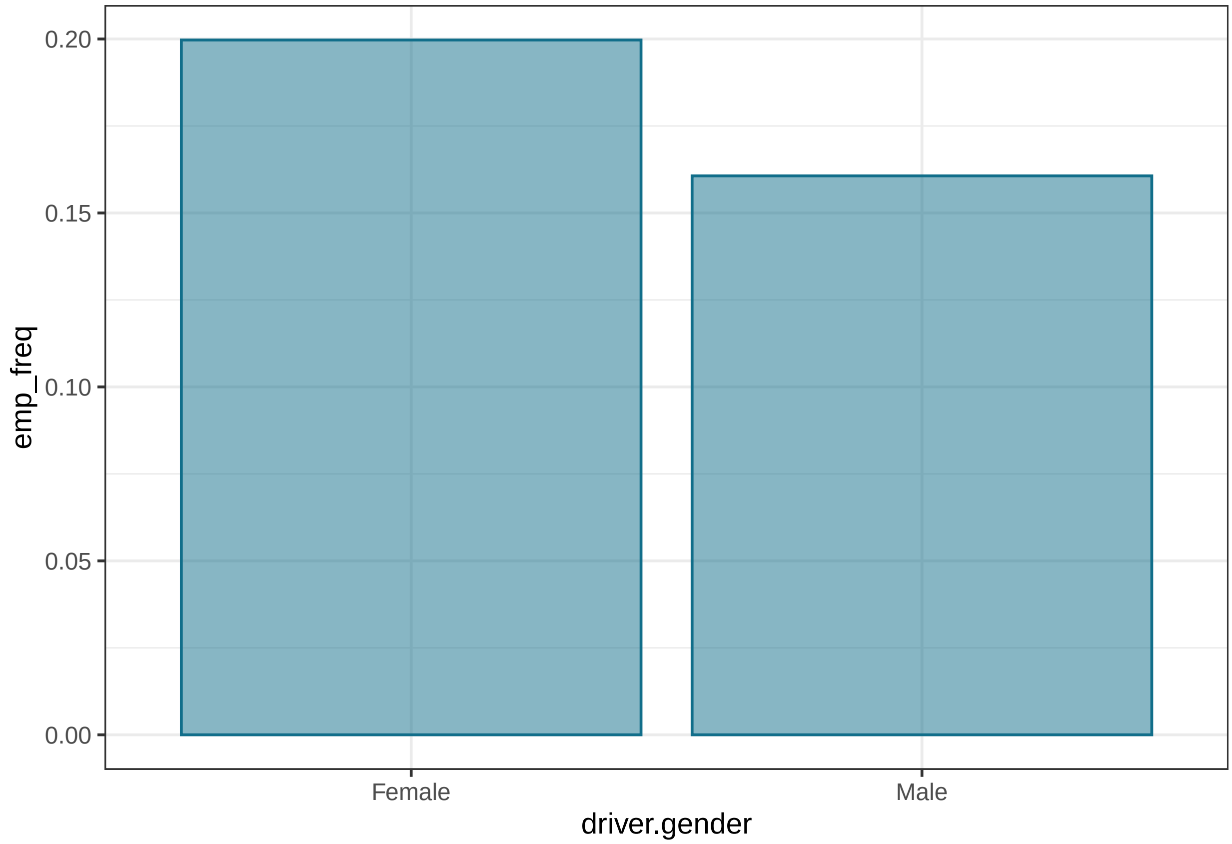

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
freq_by_gender <- dta %>%
  group_by(driver.gender) %>%
  summarize(emp_freq = sum(clm.count) / sum(exposure))
freq_by_gender
```

```
## # A tibble: 2 x 2
##   driver.gender emp_freq
##   <chr>            <dbl>
## 1 Female           0.200
## 2 Male             0.161
```

```r
ggplot(freq_by_gender, aes(x = driver.gender, y = emp_freq)) + theme_bw() +
  geom_bar(stat = "identity", col = KULbg, fill = KULbg, alpha = .5)
```



Split claim by driver gender

```r
with(dta, table(driver.gender, clm.count))
```

```
##              clm.count
## driver.gender     0     1     2     3     4     5
##        Female  4110   365    39     4     0     1
##        Male   33481  2562   186    11     1     0
```

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.1
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.2     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
freq_glm_1 <- glm(clm.count ~ driver.gender, offset = log(exposure),
                  family = poisson(link = "log"),
                  data = dta)
freq_glm_1 %>% broom::tidy()
```

```
## # A tibble: 2 x 5
##   term              estimate std.error statistic   p.value
##   <chr>                <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)         -1.61     0.0466    -34.5  1.46e-261
## 2 driver.genderMale   -0.218    0.0501     -4.34 1.41e-  5
```

```r
summary(freq_glm_1)
```

```
##
## Call:
## glm(formula = clm.count ~ driver.gender, family = poisson(link = "log"),
##     data = dta, offset = log(exposure))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6320  -0.4909  -0.4008  -0.2606   4.7542
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.61087    0.04663 -34.549  < 2e-16 ***
## driver.genderMale -0.21755    0.05010  -4.342 1.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 16616  on 40759  degrees of freedom
## Residual deviance: 16598  on 40758  degrees of freedom
## AIC: 23096
##
## Number of Fisher Scoring iterations: 6
```

Interpretation (cf. Charpentier, STT5100-11)

We want to model the annual frequency of claims according to the number of claims incurred during the exposure duration.

The modality Male Driver is significanlty different from Women Driver. The Coefficient -0.21 is also negative. We estimate that we should be around 20% less high in term of frequency for Male individual

there is 21% less chance to have an accident being a Male.

lambda represent my prediction for the annual frequency

lambda_1 = exp(-1.61)

lambda_1=exp(freq_glm_1$coefficients[1])lambda_2 = exp(freq_glm_1$coefficients[1]+freq_glm_1$coefficients[2]) (lambda_2-lambda_1)/lambda_1

Globally, we saw previously that the annual frequency was around 16%. If the driver is a woman, the frequency is a little higher at 20% If the driver is a man, the frequency is around 16% We are 20% lover for men than women in terms of annual frequency.

very good website for website shaping: https://environmentalcomputing.net/statistics/glms/interpret-glm-coeffs/#:~:text=In%20linear%20models%2C%20the%20interpretation,in%20altitude%20of%201%20unit.

–> HUgo package

Let's run some experiments to illustrate the effect of the smoothing parameter (sp = .), the number (k = .) and type of basis functions (bs = .). We use the mcycle data from {MASS}.

ccm: size of engine hp: horse power

```
KULbg <- "#116E8A"

# number 1
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(mgcv)
```

```
## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## This is mgcv 1.8-35. For overview type 'help("mgcv-package")'.
```

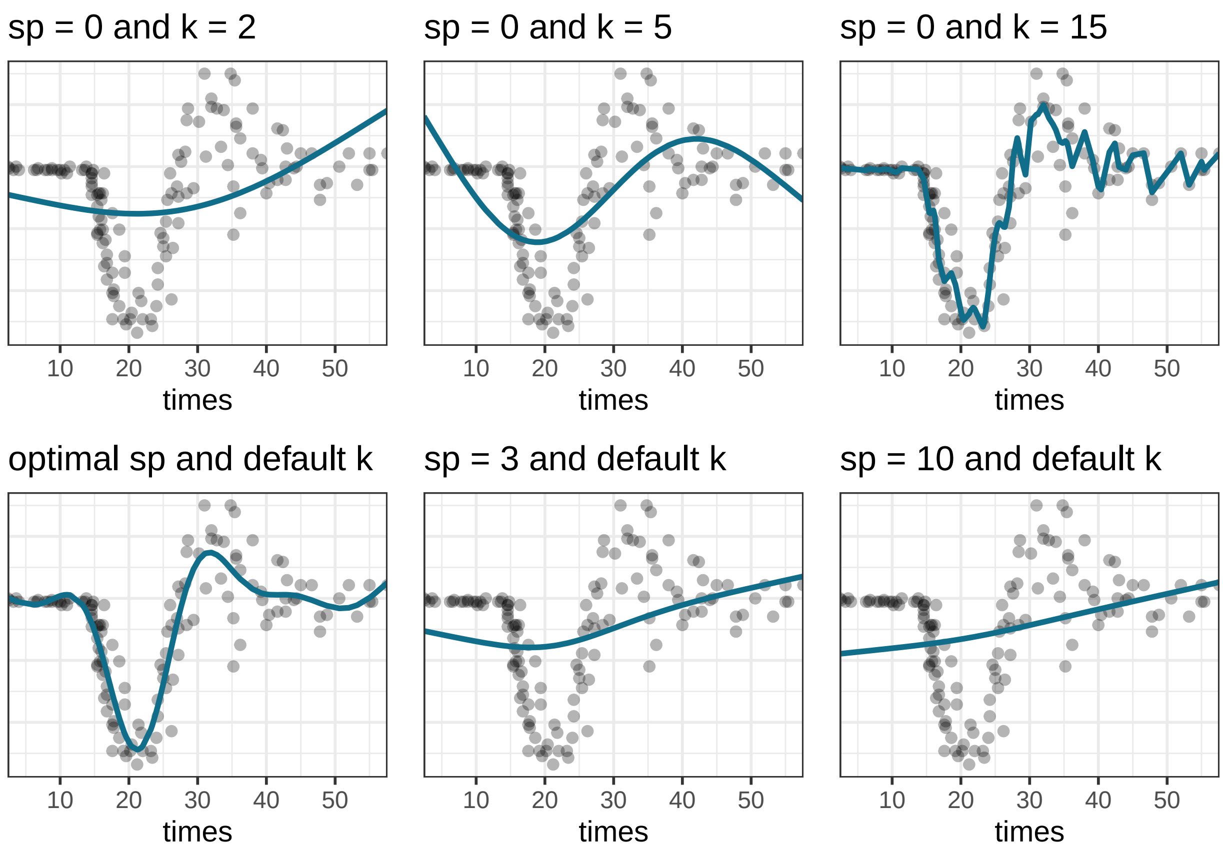
```
# In the package MASS, mcycle is dataset from a Simulated #Motorcycle accident


bias_model <- gam(accel ~ s(times, sp = 0, k = 2), data = mcycle)
```

```
## Warning in smooth.construct.tp.smooth.spec(object, dk$data, dk$knots): basis dimension, k, increased
```

```
mcycle$predictions <- predict(bias_model, mcycle)
p_1 <- ggplot(mcycle, aes(times, accel)) + theme_bw() +
  geom_point(alpha = .3) +
  geom_line(aes(times, predictions), size = 1.0, color = KULbg) +
  theme(axis.title.y = element_blank(),
```

```
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank()) +
  scale_x_continuous(expand = c(0, 0)) +
  ggtitle("sp = 0 and k = 2")

# number 2
bias_model <- gam(accel ~ s(times, sp = 0, k = 5), data = mcycle)
mcycle$predictions <- predict(bias_model, mcycle)
p_2 <- ggplot(mcycle, aes(times, accel)) + theme_bw() +
  geom_point(alpha = .3) +
  geom_line(aes(times, predictions), size = 1.0, color = KULbg) +
  theme(axis.title.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank()) +
  scale_x_continuous(expand = c(0, 0)) +
  ggtitle("sp = 0 and k = 5")
# number 3
bias_model <- gam(accel ~ s(times, sp = 0, k = 55), data = mcycle)
mcycle$predictions <- predict(bias_model, mcycle)
p_3 <- ggplot(mcycle, aes(times, accel)) + theme_bw() +
  geom_point(alpha = .3) +
  geom_line(aes(times, predictions), size = 1.0, color = KULbg) +
  theme(axis.title.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank()) +
  scale_x_continuous(expand = c(0, 0)) +
  ggtitle("sp = 0 and k = 15")
# number 4
library(MASS)
bias_model <- gam(accel ~ s(times), data = mcycle)
mcycle$predictions <- predict(bias_model, mcycle)
p_4 <- ggplot(mcycle, aes(times, accel)) + theme_bw() +
  geom_point(alpha = .3) +
  geom_line(aes(times, predictions), size = 1.0, color = KULbg) +
  theme(axis.title.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank()) +
  scale_x_continuous(expand = c(0, 0)) +
  ggtitle("optimal sp and default k")
# number 5
bias_model <- gam(accel ~ s(times, sp = 3), data = mcycle)
mcycle$predictions <- predict(bias_model, mcycle)
p_5 <- ggplot(mcycle, aes(times, accel)) + theme_bw() +
  geom_point(alpha = .3) +
  geom_line(aes(times, predictions), size = 1.0, color = KULbg) +
  theme(axis.title.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank()) +
  scale_x_continuous(expand = c(0, 0)) +
  ggtitle("sp = 3 and default k")
# number 6
bias_model <- gam(accel ~ s(times, sp = 20), data = mcycle)
mcycle$predictions <- predict(bias_model, mcycle)
```

```r
p_6 <- ggplot(mcycle, aes(times, accel)) + theme_bw() +
  geom_point(alpha = .3) +
  geom_line(aes(times, predictions), size = 1.0, color = KULbg) +
  theme(axis.title.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank()) +
  scale_x_continuous(expand = c(0, 0)) +
  ggtitle("sp = 10 and default k")

gridExtra::grid.arrange(p_1, p_2, p_3, p_4, p_5, p_6, nrow = 2)
```



Include a smooth effect of times via s(times) . sp = . specifies a value for the smoothing parameter k = . fixes the number of basis functions bs = “cr” indicates which type of basis functions shoud be used. Here “cr”refers to the cubic spline basis

```r
model <- gam(accel ~ s(times, sp = 1.2,
                       k = 5, bs = "cr"),
             family = gaussian, data = mcycle)
```

Inspection of the fitter model

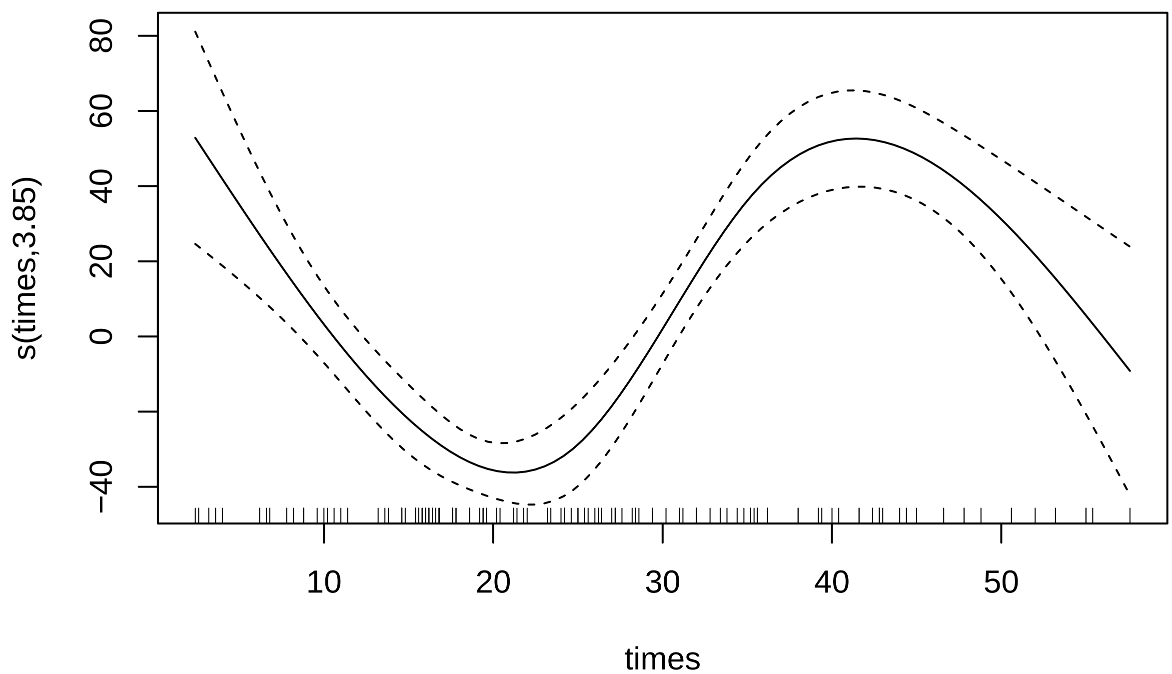
```r
print(model)
```

```
##
## Family: gaussian
```

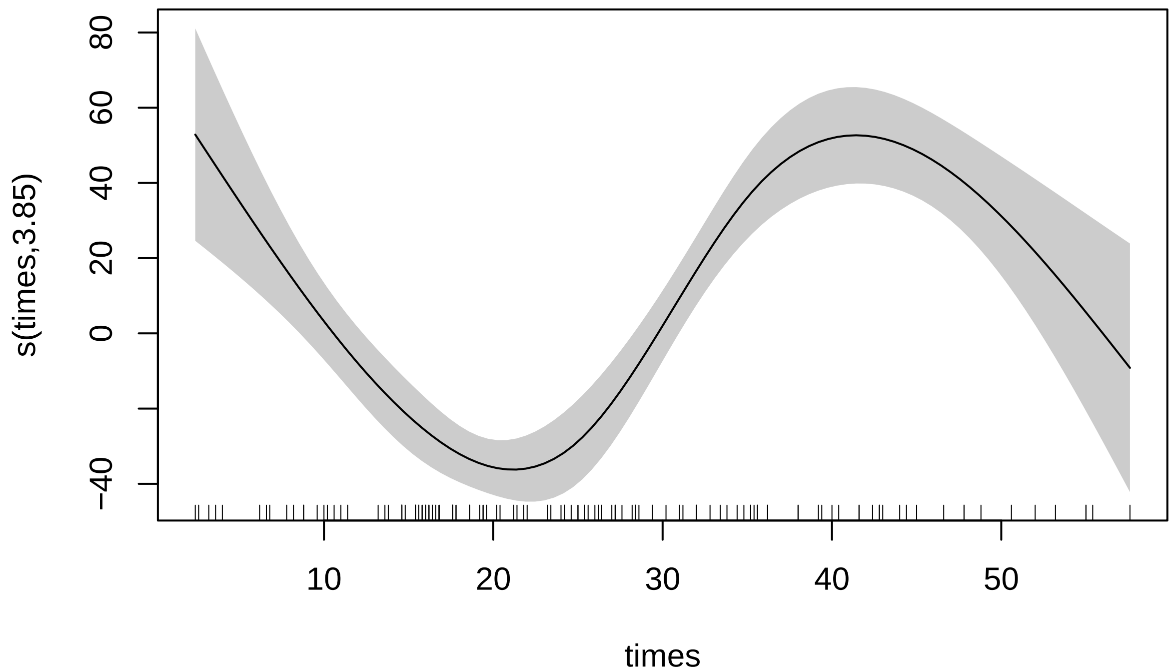
```
## Link function: identity
##
## Formula:
## accel ~ s(times, sp = 1.2, k = 5, bs = "cr")
##
## Estimated degrees of freedom:
## 3.85  total = 4.85
##
## GCV score: 1404.967
```

Visualization fo the fitted smoothers

```
plot(model, pages = 1, scheme = 0)
```



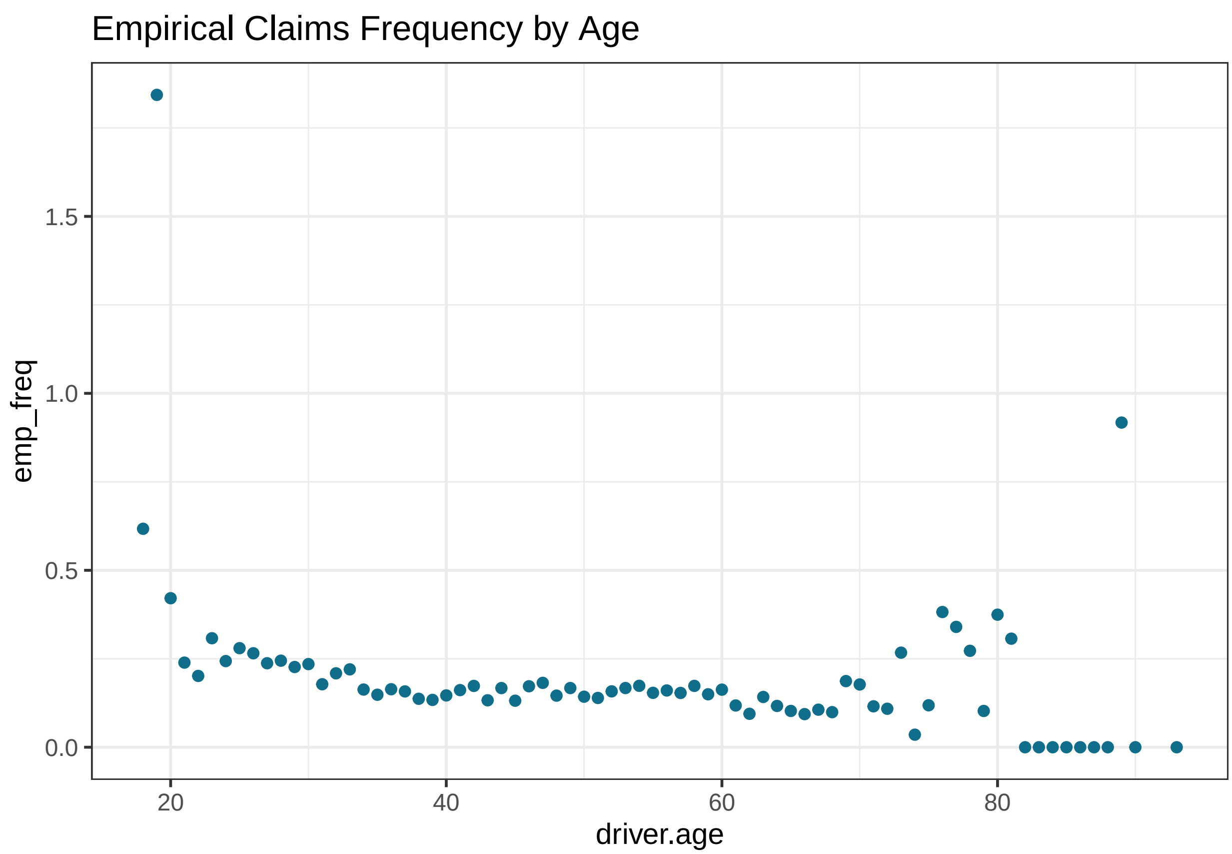

```
plot(model, pages = 1, scheme = 1)
```



Come back on the original data base

Representation of the Empirical Claims Frequency by Age

```
dta %>% group_by(driver.age) %>%
  summarize(emp_freq = sum(clm.count) / sum(exposure)) %>%
  ggplot(aes(x = driver.age, y = emp_freq)) + theme_bw() +
  geom_point(color = KULbg) + ggtitle("Empirical Claims Frequency by Age")
```



We explore 4 different model specifications

```
a <- min(dta$driver.age):max(dta$driver.age) # we make a grid of age.
```

Model 1 - Linear Effect of driver.age

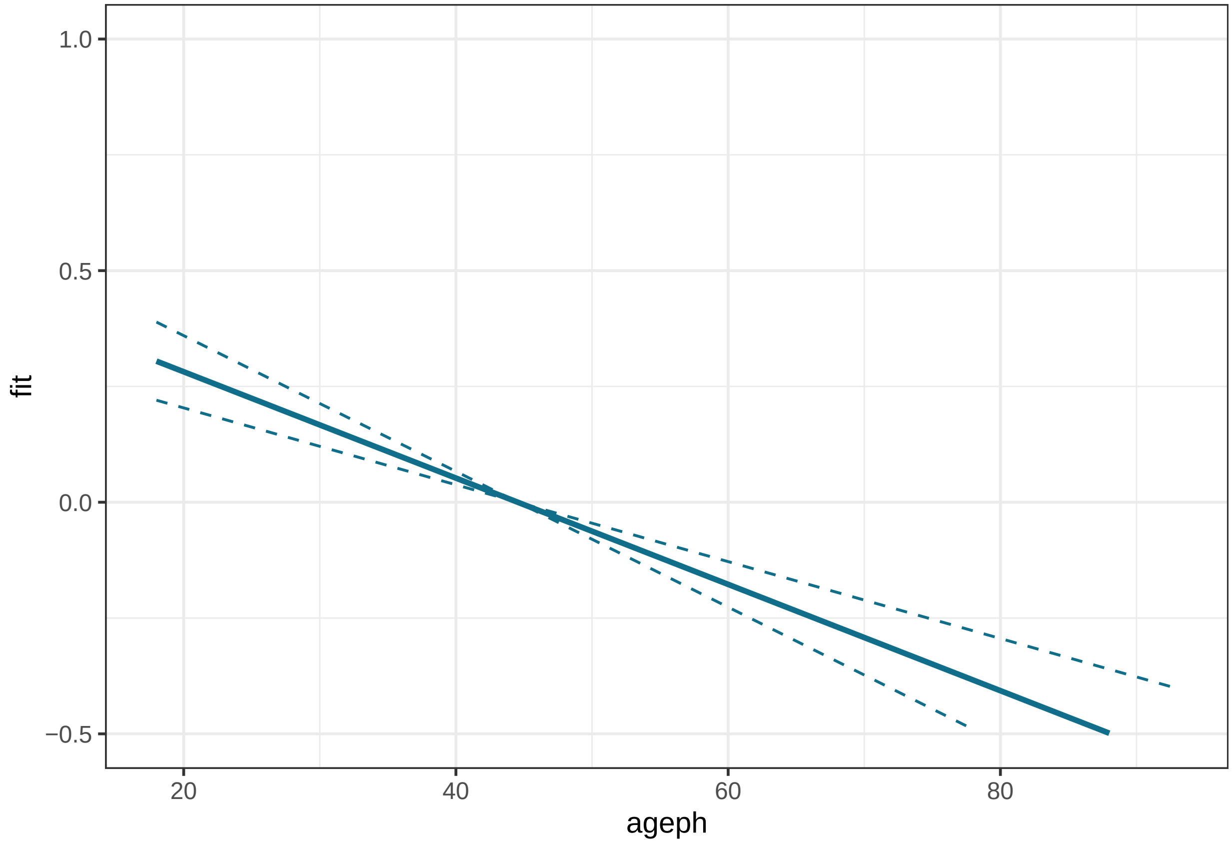
```
# Step 1: fit a model
freq_glm_age <- glm(clm.count ~ driver.age, offset = log(exposure), data = dta, family = poisson(link =

# Step2: do a prediction
pred_glm_age <- predict(freq_glm_age, newdata = data.frame(driver.age = a, exposure = 1), type = "terms

# Step3: Calculate IC for the prediction and store in a dataframe
b_glm_age <- pred_glm_age$fit
l_glm_age <- pred_glm_age$fit - qnorm(0.975)*pred_glm_age$se.fit
u_glm_age <- pred_glm_age$fit + qnorm(0.975)*pred_glm_age$se.fit

#
df <- data.frame(a, b_glm_age, l_glm_age, u_glm_age)
```

“Naive” model

```
# Visual
p_glm_age <- ggplot(df, aes(x = a)) + ylim(-0.5, 1)
p_glm_age <- p_glm_age + geom_line(aes(a, b_glm_age), size = 1, col = KULbg)
```

```r
p_glm_age <- p_glm_age + geom_line(aes(a, u_glm_age), size = 0.5, linetype = 2, col = KULbg) + geom_line
p_glm_age <- p_glm_age + xlab("ageph") + ylab("fit") + theme_bw()
p_glm_age
```

```
## Warning: Removed 5 row(s) containing missing values (geom_path).

## Warning: Removed 15 row(s) containing missing values (geom_path).
```



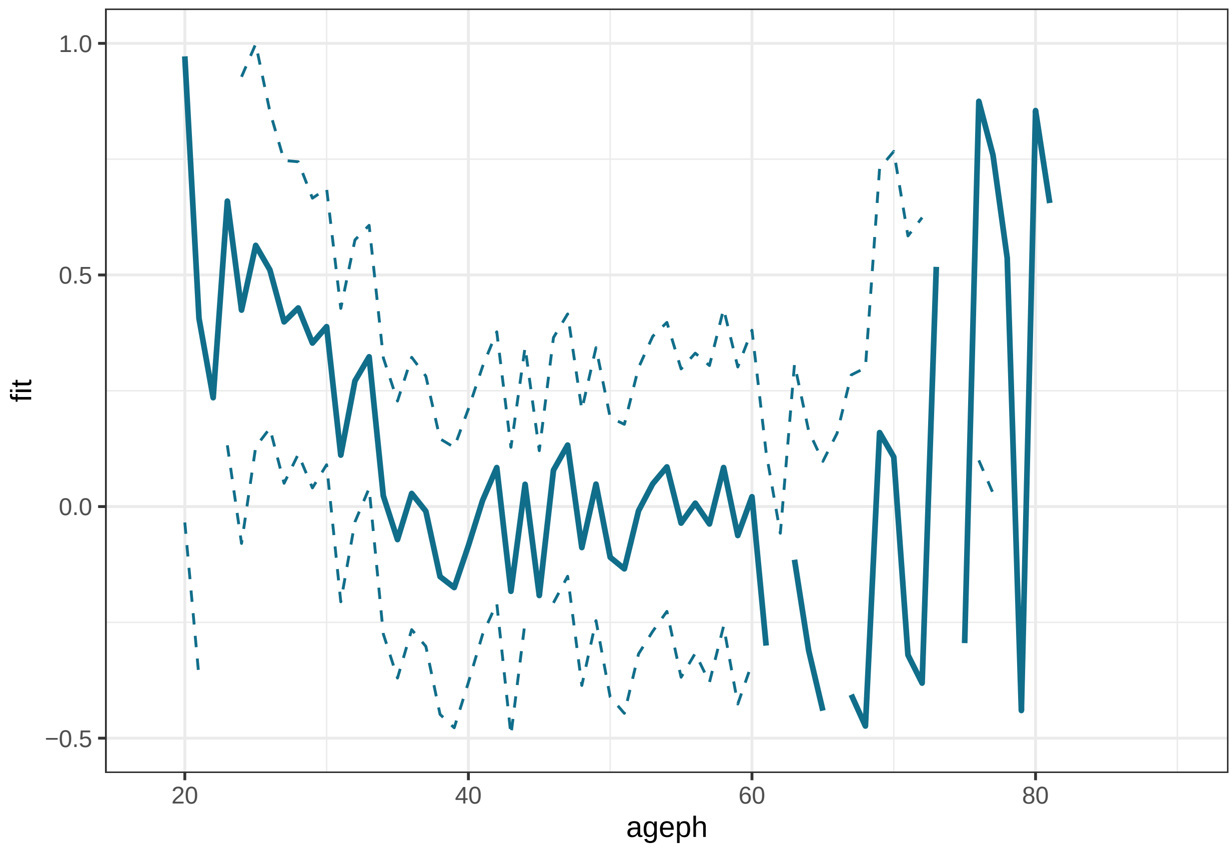
Model 2 - driver.age as a factor variable in a GLM

```r
freq_glm_age_f <- glm(clm.count ~ as.factor(driver.age), offset = log(exposure), data = dta, family = po

# Need to remove the ages 91 and 93 from a
a_bis <- a[1:73]

pred_glm_age_f <- predict(freq_glm_age_f, newdata = data.frame(driver.age = a_bis, exposure = 1), type =

b_glm_age_f <- pred_glm_age_f$fit
l_glm_age_f <- pred_glm_age_f$fit -
  qnorm(0.975)*pred_glm_age_f$se.fit
u_glm_age_f <- pred_glm_age_f$fit +
  qnorm(0.975)*pred_glm_age_f$se.fit

df <- data.frame(a_bis, b_glm_age_f,
                 l_glm_age_f, u_glm_age_f)
```

We have a very wiggely outcome

```
p_glm_age_f <- ggplot(df, aes(x = a_bis)) + ylim(-0.5, 1)
p_glm_age_f <- p_glm_age_f + geom_line(aes(a_bis, b_glm_age_f), size = 1, col = KULbg)
p_glm_age_f <- p_glm_age_f + geom_line(aes(a_bis, u_glm_age_f), size = 0.5, linetype = 2, col = KULbg) +
p_glm_age_f <- p_glm_age_f + xlab("ageph") + ylab("fit") + theme_bw()
p_glm_age_f
```

```
## Warning: Removed 11 row(s) containing missing values (geom_path).

## Warning: Removed 22 row(s) containing missing values (geom_path).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```



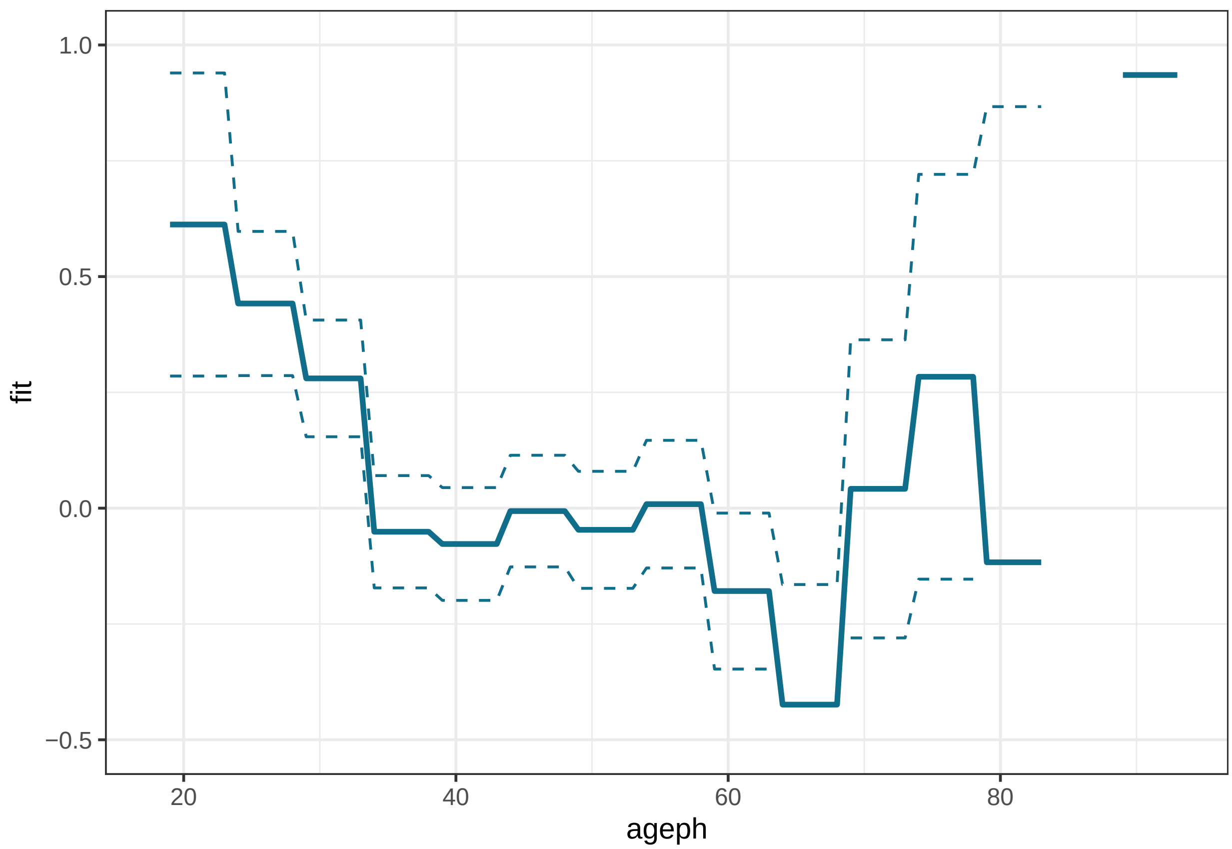
Model 3 - driver.age split into 5 years bin using a GLM

```
level <- seq(min(dta$driver.age), max(dta$driver.age), by = 5)

freq_glm_age_c <- glm(clm.count ~ cut(driver.age, level), offset = log(exposure), data = dta, family = p

pred_glm_age_c <- predict(freq_glm_age_c, newdata = data.frame(driver.age = a, exposure = 1), type = "te

b_glm_age_c <- pred_glm_age_c$fit
l_glm_age_c <- pred_glm_age_c$fit -
  qnorm(0.975)*pred_glm_age_c$se.fit
```

```
u_glm_age_c <- pred_glm_age_c$fit +
  qnorm(0.975)*pred_glm_age_c$se.fit

df <- data.frame(a, b_glm_age_c,
                 l_glm_age_c, u_glm_age_c)
```

```
p_glm_age_c <- ggplot(df, aes(x = a)) + ylim(-0.5, 1)
p_glm_age_c <- p_glm_age_c + geom_line(aes(a, b_glm_age_c), size = 1, col = KULbg)
p_glm_age_c <- p_glm_age_c + geom_line(aes(a, u_glm_age_c), size = 0.5, linetype = 2, col = KULbg) + ge
p_glm_age_c <- p_glm_age_c + xlab("ageph") + ylab("fit") + theme_bw()
p_glm_age_c
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 11 row(s) containing missing values (geom_path).

## Warning: Removed 16 row(s) containing missing values (geom_path).
```



Model 4: we use a smoothing effect of driver.age

```
freq_gam_age <- gam(clm.count ~ s(driver.age),
                    offset = log(exposure),
                    data = dta,
```

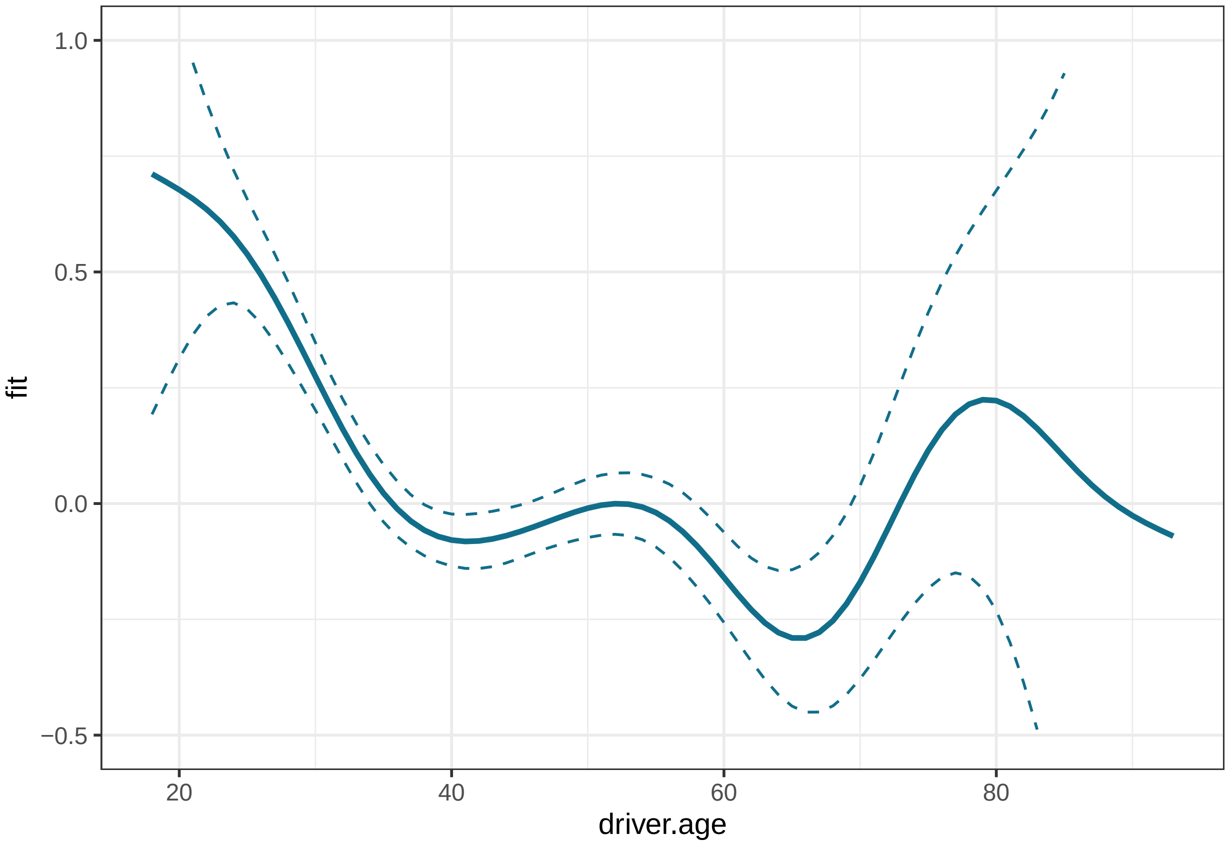
```r
                    family = poisson(link = "log"))

pred_gam_age <- predict(freq_gam_age,
                        newdata = data.frame(driver.age = a, exposure = 1),
                        type = "terms", se.fit = TRUE)

b_gam_age <- pred_gam_age$fit
l_gam_age <- pred_gam_age$fit -
  qnorm(0.975)*pred_gam_age$se.fit
u_gam_age <- pred_gam_age$fit +
  qnorm(0.975)*pred_gam_age$se.fit
df <- data.frame(a, b_gam_age,
                 l_gam_age, u_gam_age)

summary(freq_gam_age)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## clm.count ~ s(driver.age)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.81613    0.01729    -105   <2e-16 ***
## ---
## 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(driver.age) 7.18  7.947  113.4  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.027   Deviance explained = 0.64%
## UBRE = -0.59455  Scale est. = 1         n = 40760
```

```r
# we want to capture the smoothing effect of age
p_gam_age <- ggplot(df, aes(x = a)) + ylim(-0.5, 1)
p_gam_age <- p_gam_age + geom_line(aes(a, b_gam_age), size = 1, col = KULbg)
p_gam_age <- p_gam_age + geom_line(aes(a, u_gam_age), size = 0.5, linetype = 2, col = KULbg) + geom_line
p_gam_age <- p_gam_age + xlab("driver.age") + ylab("fit") + theme_bw()
p_gam_age
```

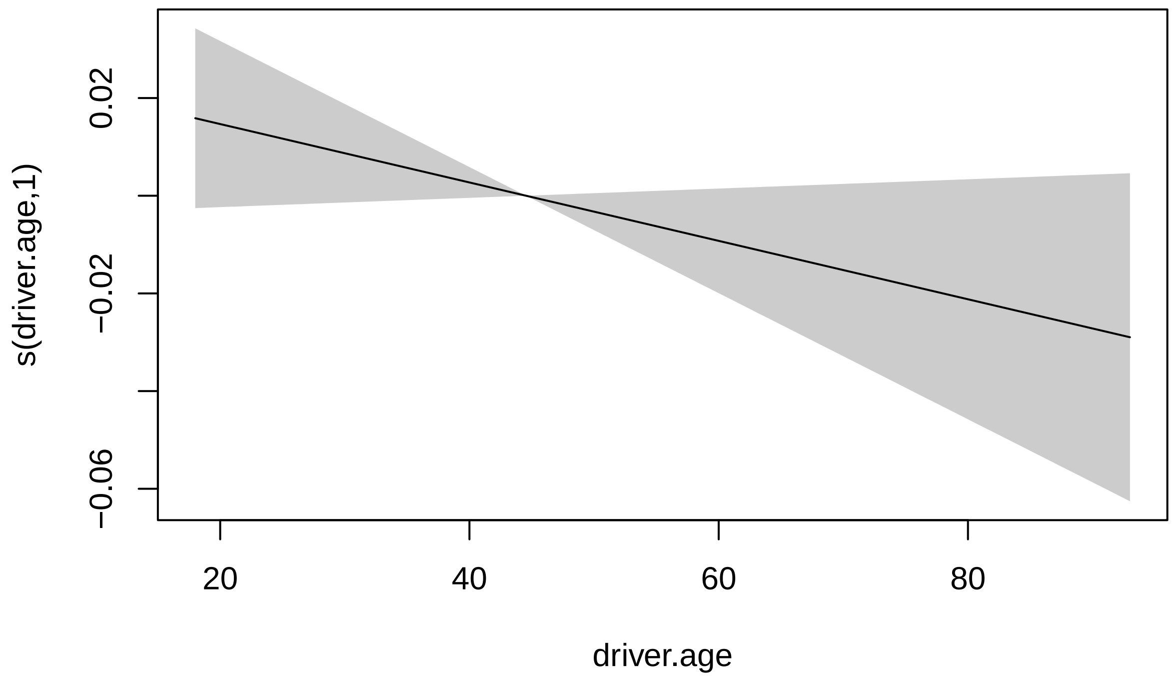
```
## Warning: Removed 11 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 10 row(s) containing missing values (geom_path).
```



Interesting shape: age.driver has an effect on f(age.driver) which decrease from the youg age to 40yo. Slight increase, decrease again and goes up from 65 to 80. Experienced driver tend to have less accident until a certain age at which the reflex, hability starts decreasing.

```
library(mgcv)
freq_gam <- gam(clm.count ~ s(driver.age), offset = log(exposure), familly = poisson(link = "log"), data

plot(freq_gam, scheme=1)
```



Examination of a model

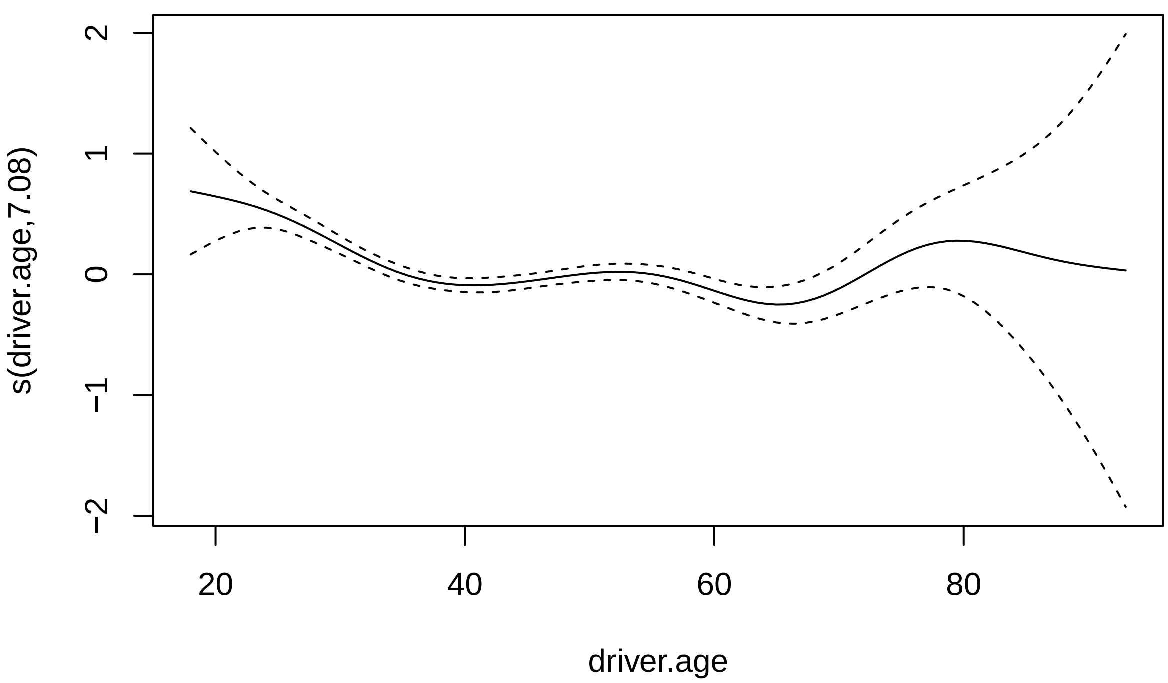
```
freq_gam_1 <- gam(clm.count  ~
                  driver.gender + fuel.type + vehicle.age +
                  s(driver.age),

                offset = log(exposure),
                data = dta,
                family = poisson(link = "log"))
```

```
summary(freq_gam_1)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## clm.count ~ driver.gender + fuel.type + vehicle.age + s(driver.age)
##
## Parametric coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.465027   0.051367 -28.521  < 2e-16 ***
## driver.genderMale -0.201817   0.050185  -4.021 5.78e-05 ***
## fuel.typeGasoline  0.043386   0.142893   0.304    0.761
## fuel.typeLPG       0.130044   0.219211   0.593    0.553
```

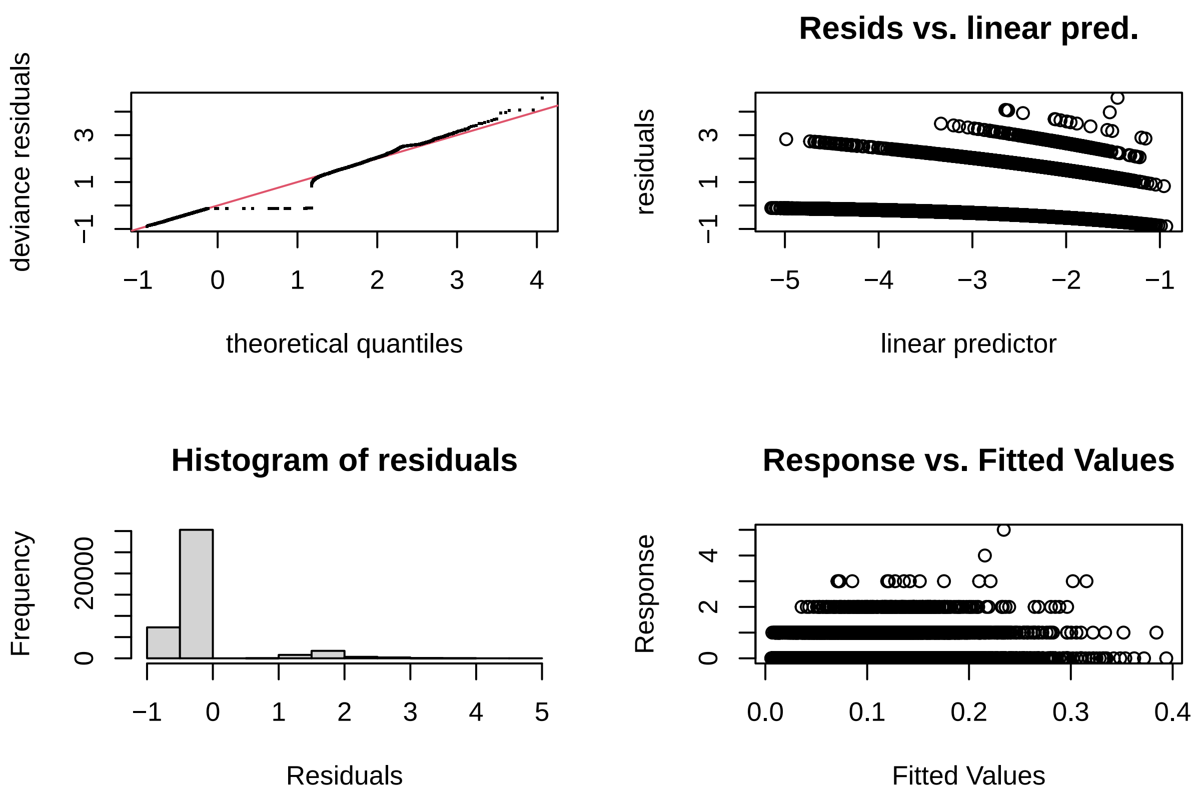
```
## vehicle.age       -0.056993   0.007165  -7.955 1.79e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                  edf Ref.df Chi.sq p-value
## s(driver.age) 7.081  7.881   93.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0287   Deviance explained = 1.13%
## UBRE = -0.59635  Scale est. = 1         n = 40760
```

```r
plot(freq_gam_1, select = 1)
```



str(dta)

```r
gam.check(freq_gam_1)
```




```
##
## Method: UBRE   Optimizer: outer newton
## full convergence after 2 iterations.
## Gradient range [1.174645e-07,1.174645e-07]
## (score -0.5963467 & scale 1).
## Hessian positive definite, eigenvalue range [9.019088e-06,9.019088e-06].
## Model rank =  14 / 14
##
## 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(driver.age) 9.00 7.08    0.85   0.095 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model with age and year licenced Had to set a lower k to avoid error For more info, cf: https://stackoverflow.com/questions/62816900/gams-in-r-fewer-unique-covariate-combinations-than-df https://stat.ethz.ch/pipermail/r-sig-ecology/2011-May/002148.html

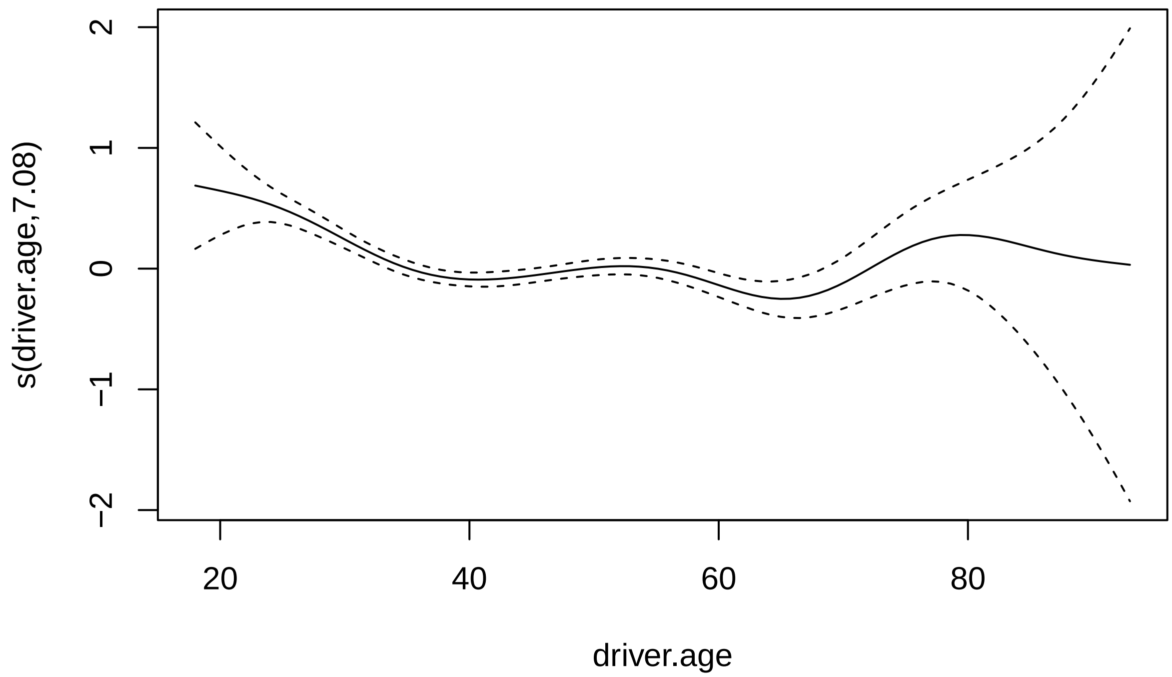
```
freq_gam_2 <- gam(clm.count  ~
                  driver.gender + fuel.type + vehicle.age +
                  s(driver.age) + s(yrs.licensed, k = 8),

                offset = log(exposure),
```

```r
                 data = dta,
                 family = poisson(link = "log"))
```

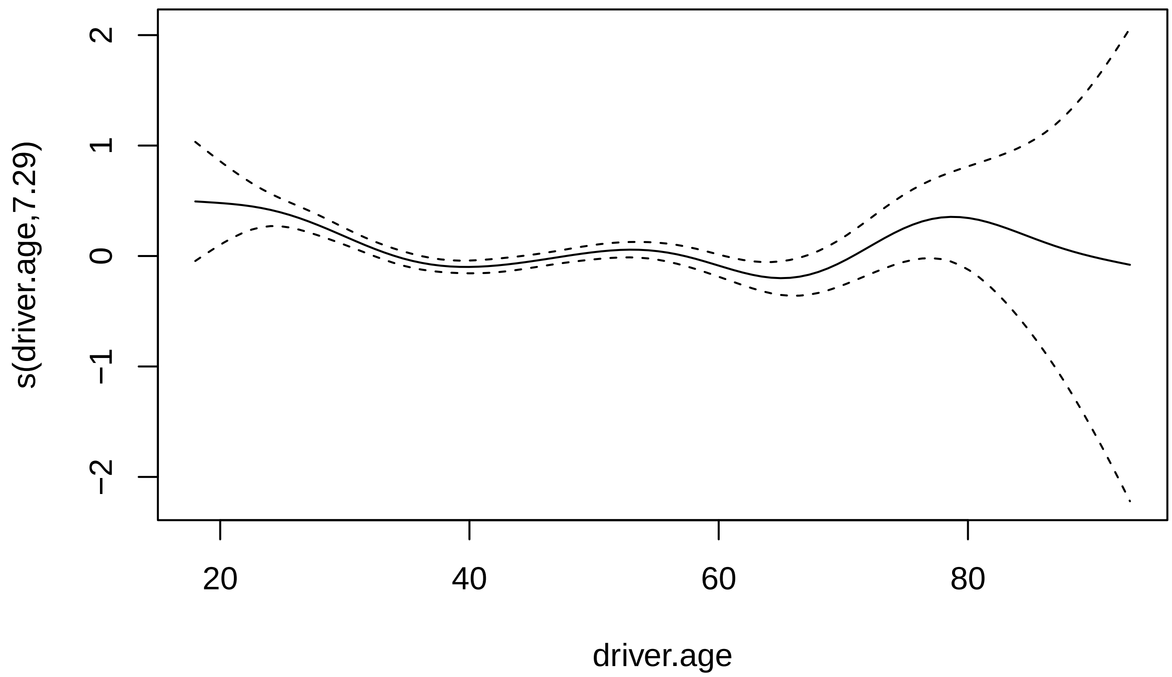
```r
summary(freq_gam_2)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## clm.count ~ driver.gender + fuel.type + vehicle.age + s(driver.age) +
##     s(yrs.licensed, k = 8)
##
## Parametric coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.576653   0.052530 -30.014  < 2e-16 ***
## driver.genderMale -0.180109   0.050239  -3.585 0.000337 ***
## fuel.typeGasoline -0.012969   0.143006  -0.091 0.927738
## fuel.typeLPG       0.114881   0.219267   0.524 0.600324
## vehicle.age       -0.036296   0.007343  -4.943 7.71e-07 ***
## ---
## 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(driver.age)   7.291  8.029   60.6  <2e-16 ***
## s(yrs.licensed) 4.963  5.813  135.7  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0326   Deviance explained = 1.99%
## UBRE = -0.59961  Scale est. = 1         n = 40760
```

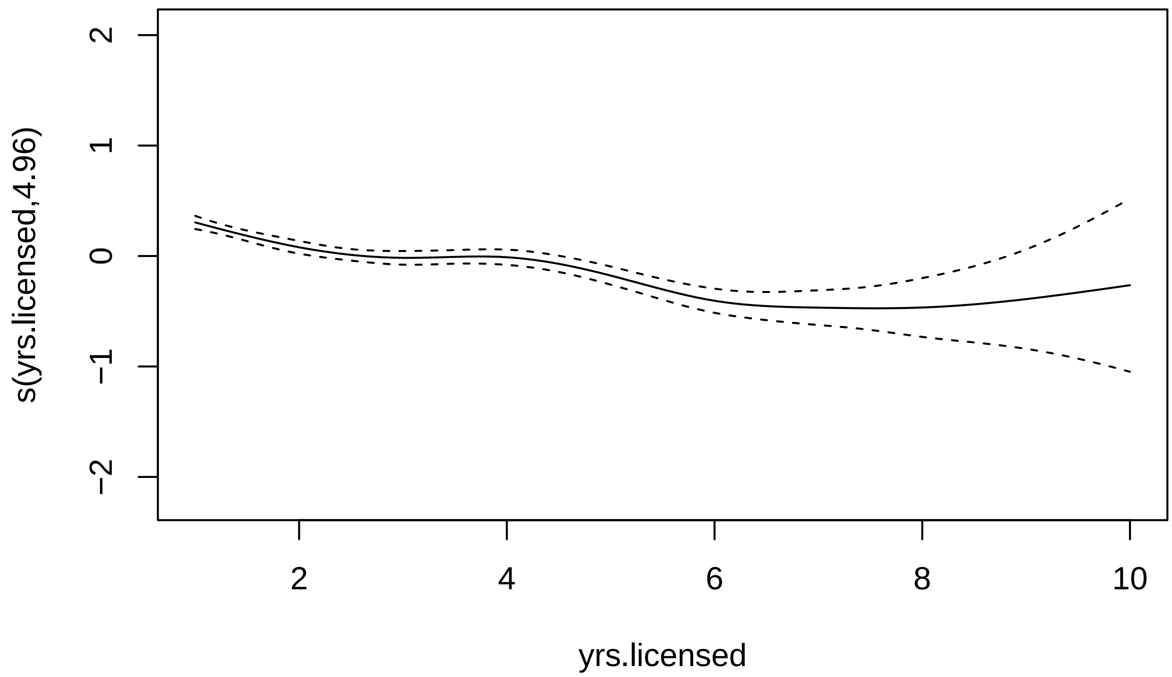
```r
plot(freq_gam_1, select = 1)
```




```
plot(freq_gam_2, select = 1)
```




```
plot(freq_gam_2, select = 2)
```



The precedent information is strengthened by the results using the number of years licenced: Experienced drivers tend to have less accident.