Nonparametric Statistics - Exam Like Lab Ex II

Noé Debrois, Person Code 10949145, ID 242751

2024-02-01

Exercise 1

Dr. Bisacciny, Ph.D is becoming increasingly worried about the fact that the bees that he decided to place near a corn field close to the Italian city of Milan tend to die way earlier than the ones that are placed close to the small village of Jovençan, in Aosta Valley. He suspects that additional factors that influence the survival time of a the bee are its productivity, and its weight. For this reason, he decides to run an experiment, in which he selects 10,000 bees, placing 5,000 of them in Aosta Valley and 5,000 in Milan. Dr. Bisacciny runs the experiment for 50 days, during which he annotates when a bee passes away. After the end of the experiment, he starts to analyse the data. In the file ex02.rda you can find a dataframe with information about the status of the bee (1 if alive, 2 if dead), the survival time of the bee (surv.time), its weight (weight) and productivity (prod) as well as its being in Jovençan or in Milan. Modeling this data as i.i.d. realizations of a four dimensional random variable:

Question 1.1

Build an additive model for log(surv.time), where log() is the natural logarithm, using cubic b-spline terms for the main effects, a dummy term for location and no interactions. After having written in proper mathematical terms the additive model you have estimated, report the adjusted R2 and the p-values of the tests. Comment the result (assume the residuals to be normal).

Synthetic description of assumptions, methods, and algorithms

What is a GAM ? It is a model whose analytical expression is : $y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + ... + f_p(x_{ip}) + \varepsilon_i$. It is called "additive model" because we calculate a separate f_j for each X_j and then we add together all of their contributions. The main features of a GAM are the following :

- The non-linear effects are modelled via the f_j ;
- Additivity allows us to interpret the model as with a standard linear regression model.

To implement this, we use the gam() function from the mgcv package.

Info about what we write inside the gam() function :

What s() (smoothing spline fit) does is basically building a "smooth" term for each of the covariates I am putting in. The behavior is very similar to smooth.spline we have seen last time, no need to set the number of knots, nor (like in the gam package) the equivalent degrees of freedom ; mgcv will take care of everything for you. Some more details here :

- With "bs" we indicate the B-spline penalised smoothing basis to use.
- "cr" provides a cubic spline basis defined by a modest sized set of knots spread evenly through the covariate values. They are penalized by the conventional integrated square second derivative cubic spline penalty. The fitting is based on penalized likelihood, where the term to be penalized is the second derivatives of the smooths.

For more info, run "?smooth.terms".

Results and brief discussion

First, let's replace the surv.time by the log(surv.time) :

Write the additive model in proper mathematical terms : $log(surv.time) = \beta_0 + f(weight) + f(prod) + \beta_1 location + \varepsilon$

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## bee$surv.time ~ s(weight, bs = "cr") + s(prod, bs = "cr") + location
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.01120    0.01421  211.90   <2e-16 ***
## locationMilan -1.84334    0.02010  -91.72   <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(weight) 1.764  2.240 2.178   0.100
## s(prod)   3.402  4.264 0.957   0.425
##
## R-sq.(adj) =  0.457   Deviance explained = 45.7%
## GCV = 1.0103  Scale est. = 1.0096    n = 10000
```

For the interpretation of the summary table, please see the other exam like exercise about regression. R^2 :

```
## [1] 0.4569872
```

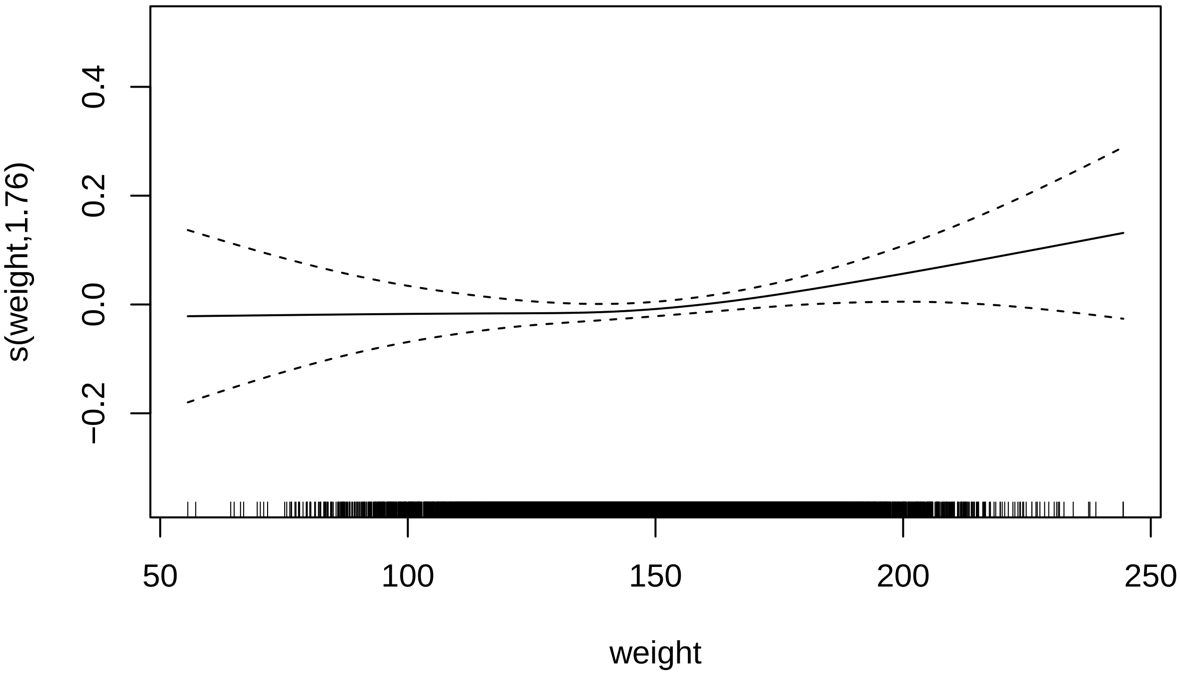
P-values of the tests (only for parametric coefficients) :

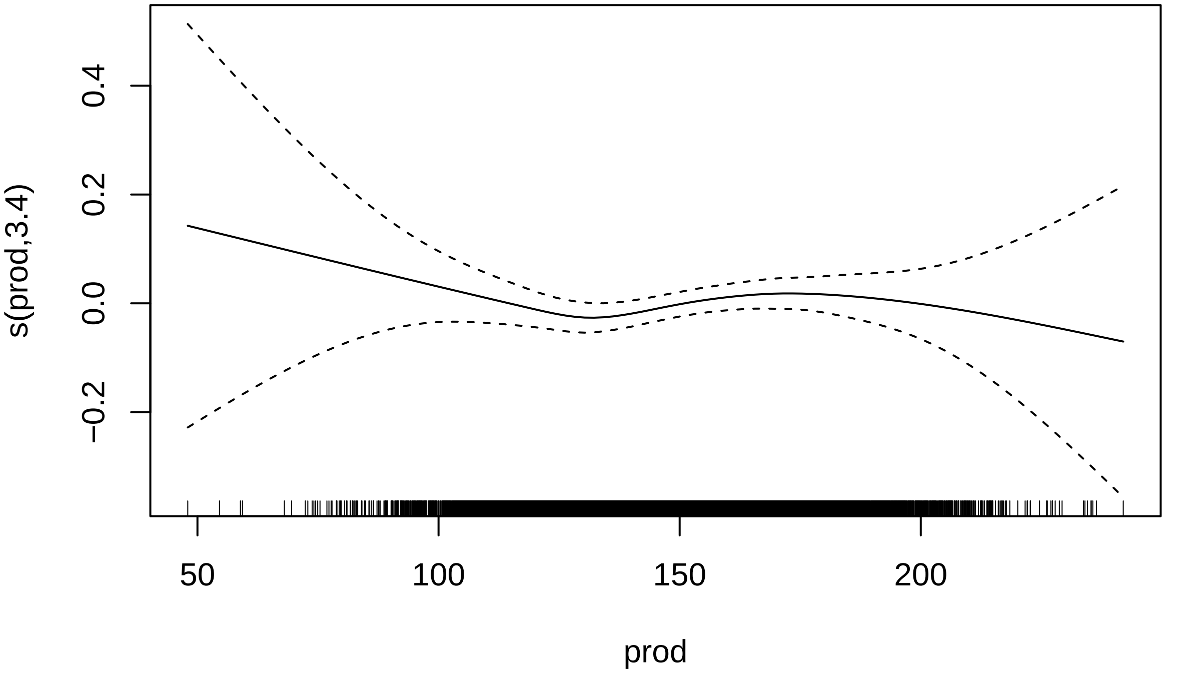
```
##   (Intercept) locationMilan
##             0             0
```

P-values of the tests (only for smooth terms) :

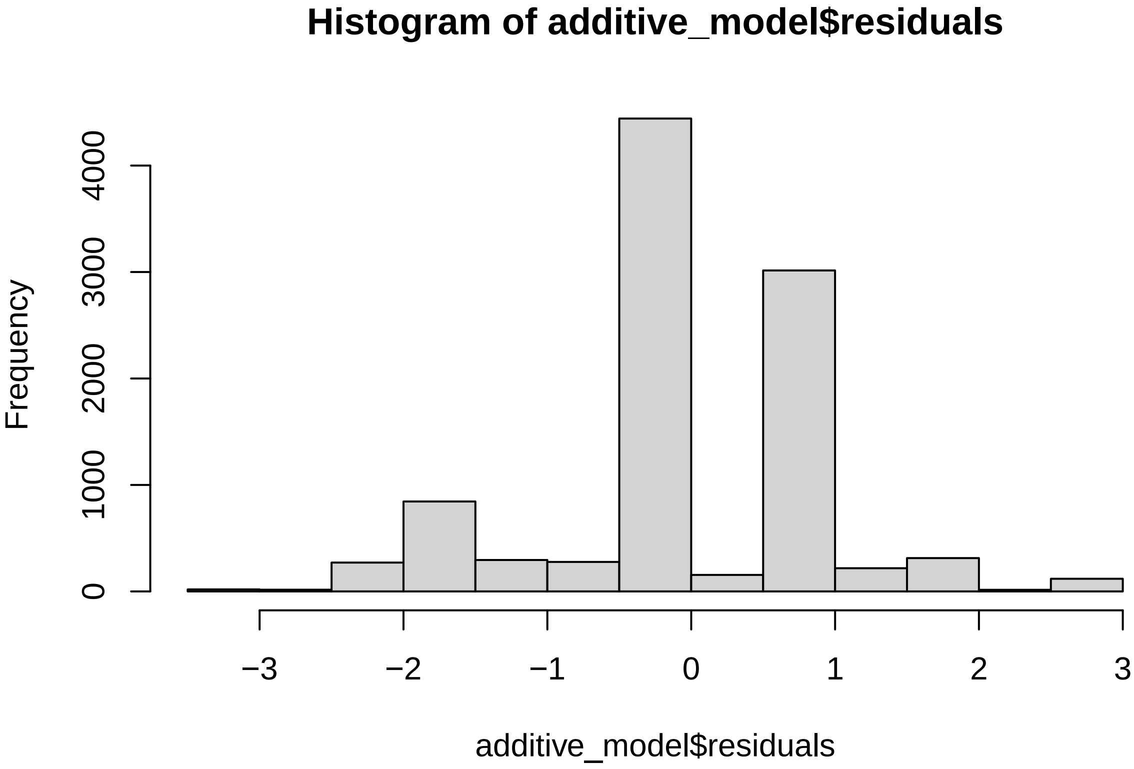
```
## [1] 0.1002523 0.4254312
```

Plot :





Plot of the residuals (assumed to be gaussian) :



Question 1.2

After having reduced the model to its significant terms, use this model to provide a pointwise prediction for the average log-survival time for a bee that lives in Milan or in Jovençan. Is it necessary to use the GAM machinery to estimate the reduced model?

Synthetic description of assumptions, methods, and algorithms

Results and brief discussion

According to the previous summary, the only significant term is the location !

In fact, I can use a simple linear model with just a term: location. No need to use the gam machinery anymore. So let's do this :

```
##
## Call:
## lm(formula = bee$surv.time ~ location, data = bee)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.01089 -0.47502 -0.06955  0.86031  2.70304
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.01089    0.01421  211.81   <2e-16 ***
## locationMilan -1.84273    0.02010  -91.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.005 on 9998 degrees of freedom
## Multiple R-squared:  0.4566, Adjusted R-squared:  0.4566
## F-statistic:  8403 on 1 and 9998 DF,  p-value: < 2.2e-16
```

All the coefficients are significant (it was expected). Here we get the average log survival time for both Milan and Jovencan using functional programming :

```
## Jovencan    Milan
## 3.010891 1.168164
```

Or, here, using classical programming :

```
## [1] 1.168164
```

```
## [1] 3.010891
```

Clearly, bees die sooner in Milan than in Jovencan.

PS : intercept = 3.01 and locationMilan = 1.16 so look directly at the coefficients of the reduced linear model…