

# Modelling Maize Project: cob\_weight.num

Modelling with smooth effect

Author: Nisia Trisconi | Zurich Data Scientists  
Reviewer: Dr. Luisa Barbanti | Zurich Data Scientists

October 30, 2023

## Contents

<b>1</b>	<b>Freezing Package versions</b>	<b>2</b>
<b>2</b>	<b>Load packages</b>	<b>2</b>
<b>3</b>	<b>Settings</b>	<b>2</b>
<b>4</b>	<b>Getting data</b>	<b>3</b>
<b>5</b>	<b>Design</b>	<b>5</b>
<b>6</b>	<b>Response variable: <i>cob_weight.num</i></b>	<b>5</b>
6.1	Aim . . . . .	5
6.2	Model fitting . . . . .	5
6.3	Model selection . . . . .	7
6.3.1	Logarithmic transformation . . . . .	7
6.3.2	Smooth terms . . . . .	11
6.4	Model checking . . . . .	16
6.4.1	Confidence intervals . . . . .	19
6.5	Contrasts . . . . .	21
<b>7</b>	<b>Methods description</b>	<b>22</b>
<b>8</b>	<b>Session information</b>	<b>24</b>

# 1 Freezing Package versions

The following code lines are commented out because the `{checkpoint}` package no longer works.

```
## (messages are omitted in this chunk)
##
# library(checkpoint)
# checkpoint(snapshot_date = "2022-11-15")
```

# 2 Load packages

```
## (messages are omitted from this chunk)
##
library(dplyr)
library(kableExtra)
library(ggplot2)
library(tibble) ## function rownames_to_column()
library(multcomp)
library(mgcv)
```

# 3 Settings

Global settings:

```
Sys.setenv(lang = "en_US")
theme_set(theme_bw())

if (!dir.exists("Prepared_data_and_models")) {
  dir.create("Prepared_data_and_models")
}
```

## 4 Getting data

```
d.maize <- readRDS(file = paste0("Prepared_data_and_models/",  
                                "d.maize_PreparedData.RDS"))
```

Overview of the data:

```
dim(d.maize)
```

```
[1] 108 33
```

```
head(d.maize)[1:min(ncol(d.maize), 30)]
```

```
# A tibble: 6 x 30  
  pot    soil    well depth seed.weight fungus date.germinated observations  
  <chr> <chr>    <chr> <dbl>    <dbl> <chr>    <chr>          <chr>  
1 A1    Bio garden a      3      30 <NA>    2022-05-11    <NA>  
2 A1    Bio garden b      5      34 <NA>    2022-05-11    <NA>  
3 A1    Bio garden c      2      35 <NA>    2022-05-09    <NA>  
4 A1    Bio garden d      1      40 <NA>    2022-05-10    <NA>  
5 A1    Bio garden e      4      46 <NA>    2022-05-11    <NA>  
6 A1    Bio garden f      6      37 <NA>    2022-05-11    <NA>  
# i 22 more variables: height_2022_07_05 <chr>, cob_weight <chr>, ...12 <dbl>,  
# pot.fac <fct>, soil.fac <fct>, well.fac <fct>, seed.weight.grams <dbl>,  
# fungus.fac <fct>, date.germinated.asDate <date>, obs.time <fct>,  
# broken <lgl>, height_2022_07_05.num <dbl>, plant.found <lgl>,  
# cob_weight.num <dbl>, germinated.in.lab <lgl>, germinated.in.field <lgl>,  
# germinated.yes <lgl>, days.to.germination <dbl>,  
# days.to.germination.censored <dbl>, seed_coord_y <dbl>, ...
```

```
str(d.maize)
```

```
tibble [108 x 33] (S3: tbl_df/tbl/data.frame)  
$ pot           : chr [1:108] "A1" "A1" "A1" "A1" ...  
$ soil          : chr [1:108] "Bio garden" "Bio garden" "Bio garden" "Bio garden" ...  
$ well         : chr [1:108] "a" "b" "c" "d" ...  
$ depth        : num [1:108] 3 5 2 1 4 6 6 4 5 1 ...  
$ seed.weight   : num [1:108] 30 34 35 40 46 37 27 16 23 22 ...  
$ fungus       : chr [1:108] NA NA NA NA ...  
$ date.germinated : chr [1:108] "2022-05-11" "2022-05-11" "2022-05-09" "2022-05-10" ...  
$ observations  : chr [1:108] NA NA NA NA ...  
$ height_2022_07_05 : chr [1:108] "217" "131" "143" "194" ...  
$ cob_weight    : chr [1:108] "117" "26" "61" "109" ...  
$ ...12        : num [1:108] NA NA NA NA NA NA NA NA NA NA ...  
$ pot.fac      : Factor w/ 18 levels "A1","A2","A3",...: 1 1 1 1 1 1 2 2 2 2 ...  
$ soil.fac     : Factor w/ 4 levels "Bio garden","Composana",...: 1 1 1 1 1 1 3 3 3 3 ..  
$ well.fac     : Factor w/ 6 levels "a","b","c","d",...: 1 2 3 4 5 6 1 2 3 4 ...  
$ seed.weight.grams : num [1:108] 0.3 0.34 0.35 0.4 0.46 0.37 0.27 0.16 0.23 0.22 ...  
$ fungus.fac   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
$ date.germinated.asDate : Date[1:108], format: "2022-05-11" "2022-05-11" ...  
$ obs.time     : Factor w/ 2 levels "morning","night": 2 2 2 2 2 2 2 2 2 2 ...  
$ broken       : logi [1:108] FALSE FALSE FALSE FALSE FALSE FALSE ...  
$ height_2022_07_05.num : num [1:108] 217 131 143 194 206 233 158 282 241 232 ...  
$ plant.found  : logi [1:108] TRUE TRUE TRUE TRUE TRUE TRUE ...  
$ cob_weight.num : num [1:108] 117 26 61 109 106 156 57 286 51 120 ...  
$ germinated.in.lab : logi [1:108] TRUE TRUE TRUE TRUE TRUE TRUE ...
```

```

$ germinated.in.field      : logi [1:108] FALSE FALSE FALSE FALSE FALSE FALSE ...
$ germinated.yes          : logi [1:108] TRUE TRUE TRUE TRUE TRUE TRUE ...
$ days.to.germination      : num [1:108] 11 11 9 10 11 11 11 11 NA 9 ...
$ days.to.germination.censored: num [1:108] 11 11 9 10 11 11 11 11 14 9 ...
$ seed_coord_y            : num [1:108] 1 1 2 2 3 3 1 1 2 2 ...
$ seed_coord_x            : num [1:108] 1 2 1 2 1 2 3 4 3 4 ...
$ position_field_x        : num [1:108] 1 1 1 1 1 1 1 1 1 1 ...
$ position_field_x_cm     : num [1:108] 50 50 50 50 50 50 50 50 50 50 ...
$ position_field_y        : int [1:108] 1 2 3 4 5 6 7 8 9 10 ...
$ position_field_y_cm     : num [1:108] 25 50 75 100 125 150 175 200 225 250 ...

```

## 5 Design

108 maize seeds are planted in 18 different pots, each with 6 wells.

Inside one pot, the same soil is used. The soils that were used are: Bio garden (4 pots), Composana (4 pots), herbs (6 pots), mixture (4 pots).

In each well, one maize seed is planted at a pre-defined depth (in cm), which is allocated randomly to the well. The maximum value for depth is 6cm and this corresponds to planting the seed directly in the coconut fiber that makes up the pot.

Wells in the same pots are allocated as follows:

```
[,1] [,2]
[1,] "e" "f"
[2,] "c" "d"
[3,] "a" "b"
```

The pots are arranged as follows on a table in the lab:

```
[,1] [,2] [,3] [,4] [,5] [,6]
[1,] "C1" "C2" "C3" "C4" "C5" "C6"
[2,] "B1" "B2" "B3" "B4" "B5" "B6"
[3,] "A1" "A2" "A3" "A4" "A5" "A6"
```

Seeds are watered for the first time on 04.30.2022 and are transferred to the field on 05.15.2022 according to the same scheme.

Some seeds are broken when planted, one seed develops a fungus.

Some seeds germinate in the lab, others in the field, while some seeds never germinate.

On 07.05.2022, the height of all maize plants is measured in cm. The plants that were not measured for time reasons receive a height value of `not measured`, while the plants that were not found and could hence not be measured present missing values.

On 09.16.2022, the weight of the cob is measured for all plants that have a cob. The variety of maize that was planted typically yields 1 cob per plant.

## 6 Response variable: *cob\_weight.num*

### 6.1 Aim

We are interested in testing whether *cob\_weight.num* is influenced by the following variables:

- Position in the field (i.e. *position\_field\_x\_cm* and *position\_field\_y\_cm*)
- Soil (variable *soil.fac*)
- Depth in soil (variable *depth*)
- Seed weight (variable *seed.weight*)

### 6.2 Model fitting

The variable *cob\_weight.num* is a continuous variable (in particular it is an amount), whose density is already well-centered. Consequently there should be no need to log transform it.

The variables *position\_field\_x\_cm* and *position\_field\_y\_cm* represent the position of seeds in the field. They are numeric variables which are introduced in the model as smooth variables, in a bi-dimensional way.

For this reason, a generalised additive model is fitted, using the `gam()` function in the `{mgcv}` package.

```
gam.cob_weight.num <- gam(cob_weight.num ~
  s(position_field_x_cm,
    position_field_y_cm) +
  soil.fac +
  depth +
  seed.weight,
  data = d.maize)
##
summary(gam.cob_weight.num)
```

Family: gaussian  
Link function: identity

Formula:

```
cob_weight.num ~ s(position_field_x_cm, position_field_y_cm) +
  soil.fac + depth + seed.weight
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	123.23521	30.28273	4.0695	0.0001193 ***
soil.facComposana	31.41914	17.62854	1.7823	0.0789182 .
soil.facherbs	21.64951	14.48008	1.4951	0.1392524
soil.facmixture	7.40843	16.77056	0.4418	0.6599925
depth	0.38139	3.32838	0.1146	0.9090919
seed.weight	-0.80983	0.93931	-0.8622	0.3914628

---

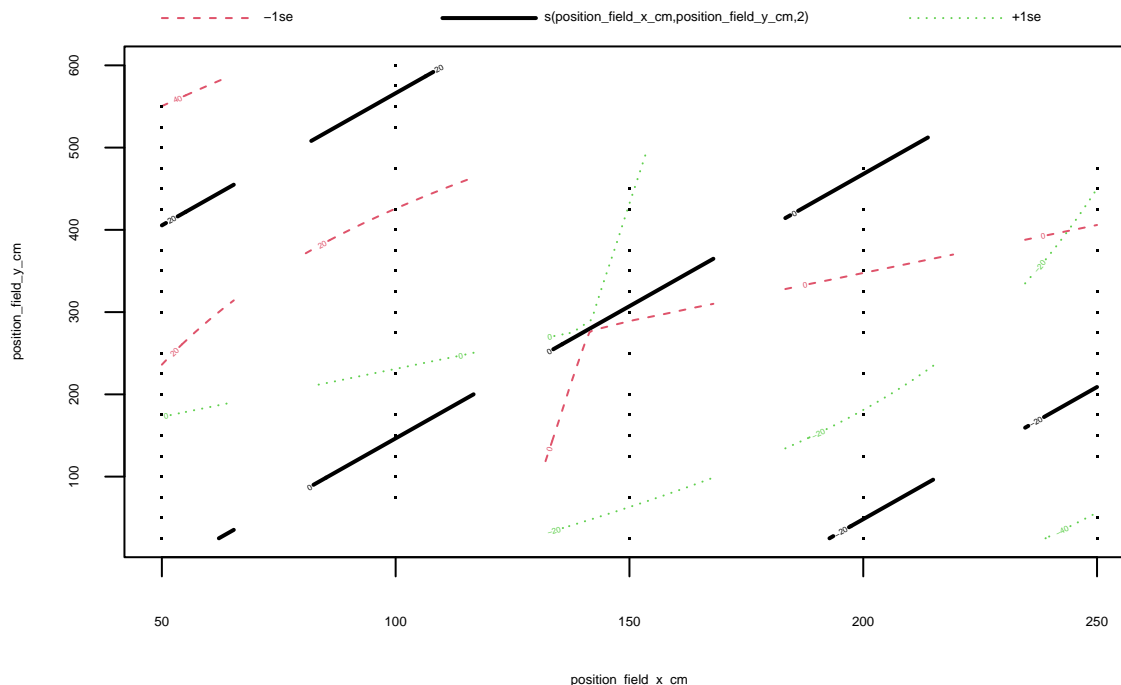
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(position_field_x_cm,position_field_y_cm)	2	2	2.337	0.1039

R-sq.(adj) = 0.0234 Deviance explained = 11%  
GCV = 2607.6 Scale est. = 2346.8 n = 80

```
plot(gam.cob_weight.num)
```



None of the explanatory variables show evidence of influencing the cob weight.

The effective degrees of freedom (referred to as EDF) are equal to 2, suggesting that there is evidence to consider introducing the two variables as linear effects. This will further be examined afterwards.

The plot displays the gradient of the estimated smooth terms, confirming a linear trend.

The explained deviance is very low, indicating that the model does not accurately represent the true distribution of the response variable.

## 6.3 Model selection

### 6.3.1 Logarithmic transformation

As introduced above, the variable *cob\_weight.num* is an amount, thus usually it is log transformed inside the model.

Since the graphical analysis did not indicate any need for applying this transformation, we are more inclined to keep the model as simple as possible. However, we fitted the model with the log-transformed response variable to see whether it could lead to an improvement, as the previous model showed evidence of being a poor fit.

```
# The response variable is amount, thus we log transform it and refit the model
gam.cob_weight.num.log <- gam(log(cob_weight.num) ~ s(position_field_x_cm,
                                                         position_field_y_cm) +
                               soil.fac +
                               depth +
                               seed.weight,
                               data = d.maize)
```

```
##
summary(gam.cob_weight.num.log)
```

Family: gaussian  
Link function: identity

Formula:  
log(cob\_weight.num) ~ s(position\_field\_x\_cm, position\_field\_y\_cm) +  
soil.fac + depth + seed.weight

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	4.3034251	0.3829933	11.2363	2.294e-16 ***
soil.facComposana	0.7217724	0.2668104	2.7052	0.008882 **
soil.facherbs	0.3681072	0.2217668	1.6599	0.102179
soil.facmixture	0.5029349	0.2712897	1.8539	0.068698 .
depth	0.0203941	0.0404187	0.5046	0.615716
seed.weight	-0.0047552	0.0117112	-0.4060	0.686164

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:

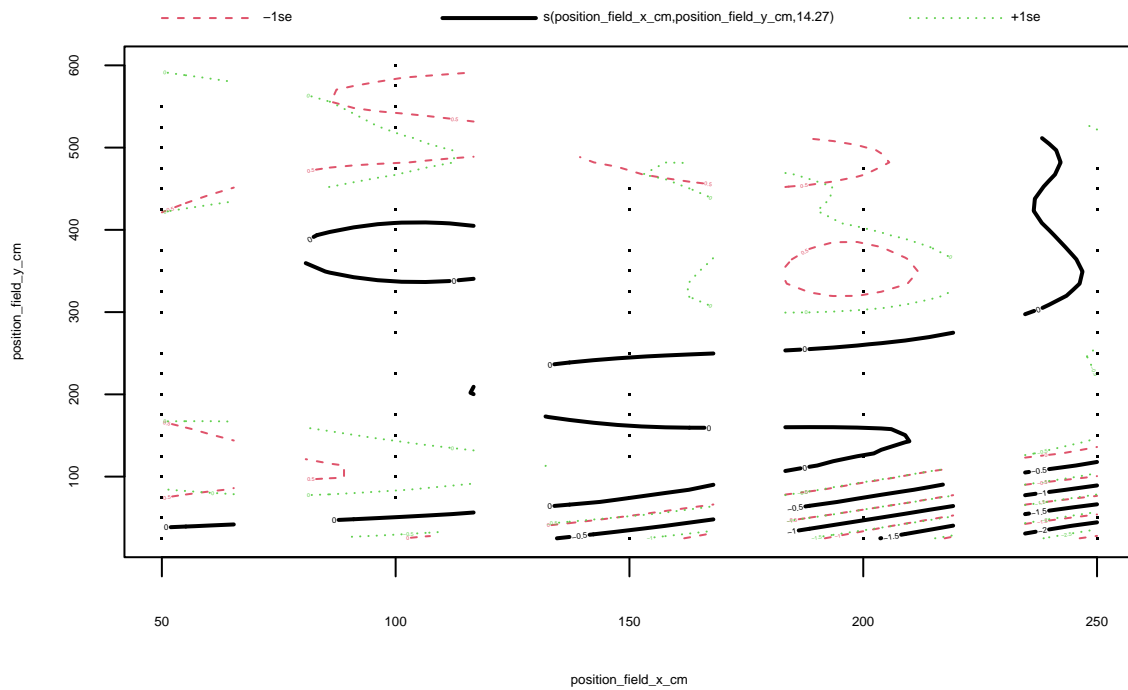
	edf	Ref.df	F	p-value
s(position_field_x_cm,position_field_y_cm)	14.266	18.869	2.9848	0.0006698 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) = 0.42 Deviance explained = 56.2%  
GCV = 0.39216 Scale est. = 0.29282 n = 80

```
plot(gam.cob_weight.num.log)
```



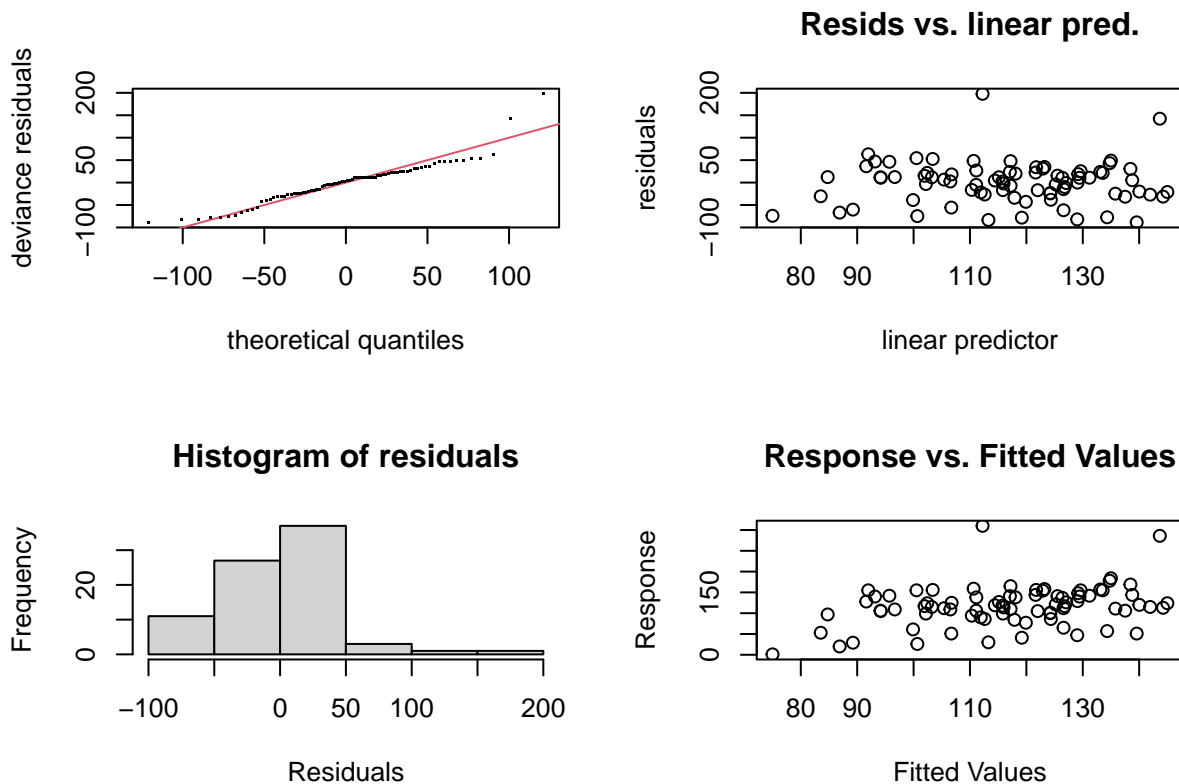
In this case, the explained deviance is higher than in the previous model. However, it's important to note that they are on two different scales, so direct comparisons based on this metric is not trustworthy.

The effective degrees of freedom (EDF) indicate a significantly higher level of complexity compared to the previous case.

As previously mentioned the two models are on different scales, thus we cannot compare them using the AIC and BIC criteria.

Our attempt to determine the better model is based on an analysis of the residual plots.

```
gam.check(gam.cob_weight.num)
```



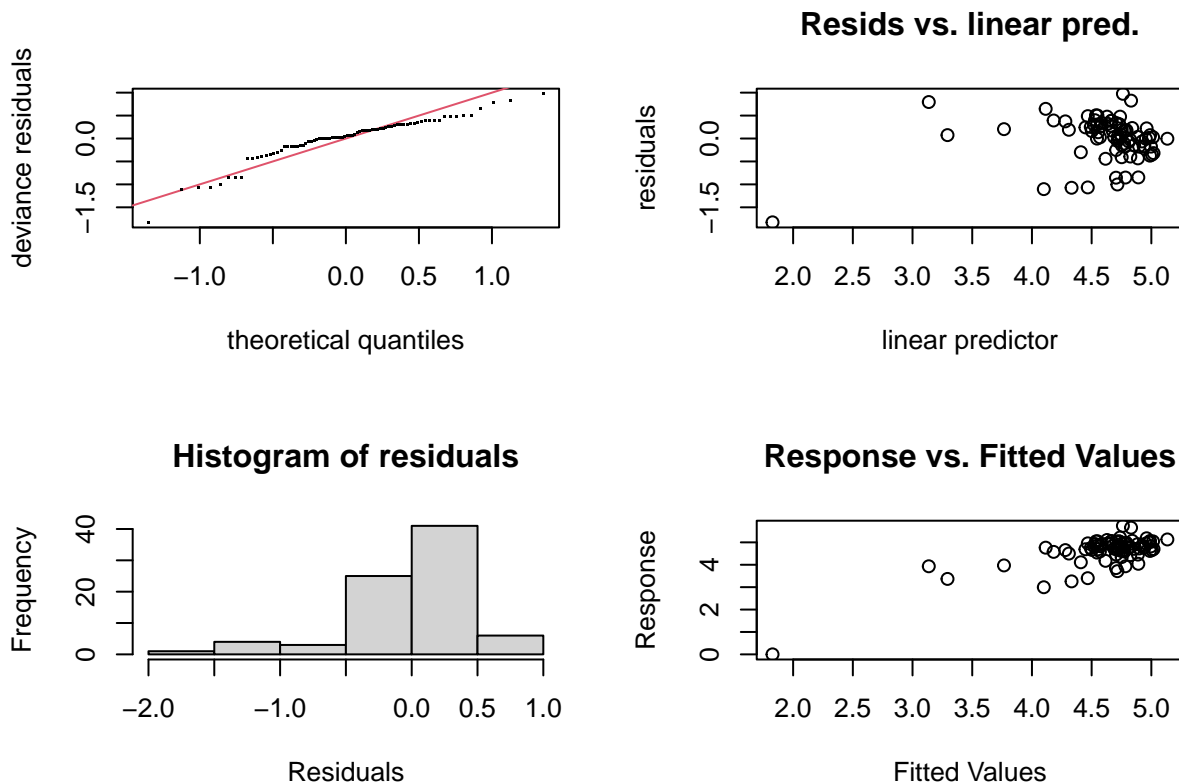
Method: GCV Optimizer: magic  
Smoothing parameter selection converged after 15 iterations.  
The RMS GCV score gradient at convergence was 0.00032639093 .  
The Hessian was positive definite.  
Model rank = 35 / 35

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(position_field_x_cm,position_field_y_cm)	29	2	1.05	0.71

The QQ-plot indicates some departure from the normality assumption, as does the histogram of the residuals.

```
gam.check(gam.cob_weight.num.log)
```



Method: GCV Optimizer: magic  
Smoothing parameter selection converged after 4 iterations.  
The RMS GCV score gradient at convergence was 5.0310027e-06 .  
The Hessian was positive definite.  
Model rank = 35 / 35

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(position_field_x_cm, position_field_y_cm)	29.0	14.3	1.13	0.92

The second model does not provide evidence of improvement over the previous model; therefore, we will retain the simpler model.

### 6.3.2 Smooth terms

As mentioned earlier, the EDF provide evidence that the smooth terms could be introduced additively and as linear terms. We will now verify this assumption by starting with their additive introduction.

We need to adjust the number of basis in the `s()` function (the function estimating the smooth term) because otherwise the model cannot be fitted; this is done by modifying `k`, which sets the upper limit on the degrees of freedom associated with the `s()` smooth.

```
k.x <- d.maize %>%
  pull(position_field_x_cm) %>%
```

```

n_distinct()
k.y <- d.maize %>%
  pull(position_field_y_cm) %>%
  n_distinct()
gam.cob_weight.num.add <- gam(cob_weight.num ~ s(position_field_x_cm, k = k.x) +
                             s(position_field_y_cm, k = k.y) +
                             soil.fac +
                             depth +
                             seed.weight,
                             data = d.maize)
summary(gam.cob_weight.num.add)

```

Family: gaussian

Link function: identity

Formula:

```

cob_weight.num ~ s(position_field_x_cm, k = k.x) + s(position_field_y_cm,
  k = k.y) + soil.fac + depth + seed.weight

```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	123.23521	30.28273	4.0695	0.0001193 ***
soil.facComposana	31.41914	17.62854	1.7823	0.0789182 .
soil.facherbs	21.64951	14.48008	1.4951	0.1392524
soil.facmixture	7.40843	16.77056	0.4418	0.6599925
depth	0.38139	3.32838	0.1146	0.9090919
seed.weight	-0.80983	0.93931	-0.8622	0.3914628

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(position_field_x_cm)	1	1	3.1364	0.08079 .
s(position_field_y_cm)	1	1	1.5291	0.22026

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

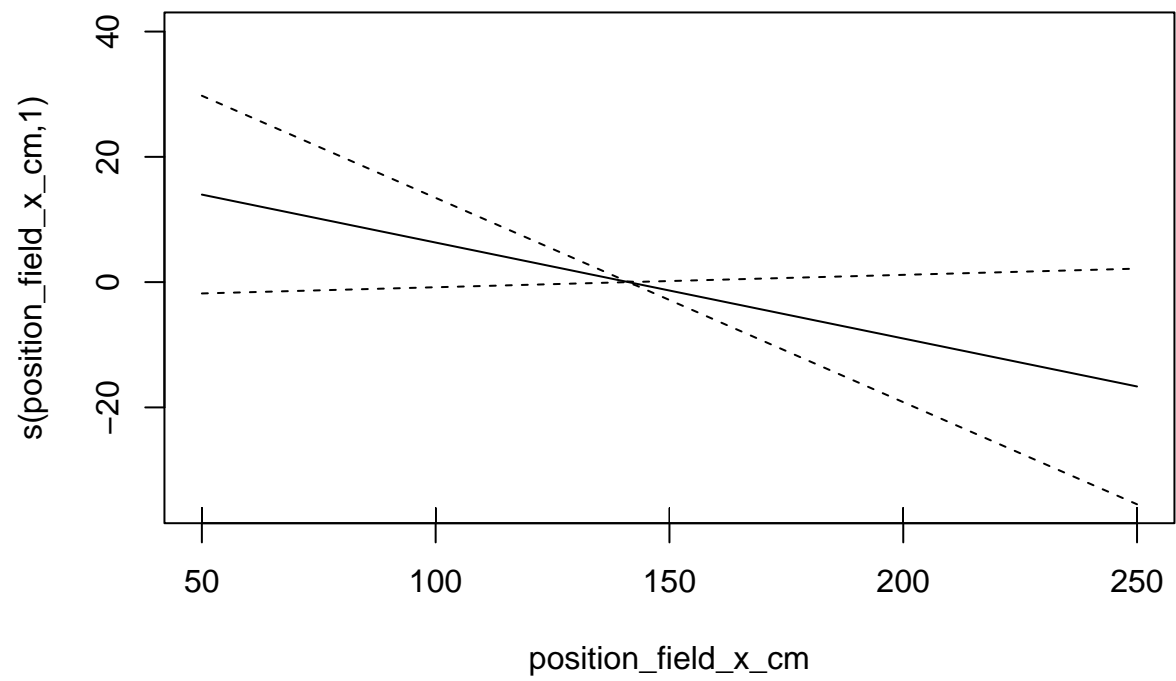
R-sq.(adj) = 0.0234 Deviance explained = 11%

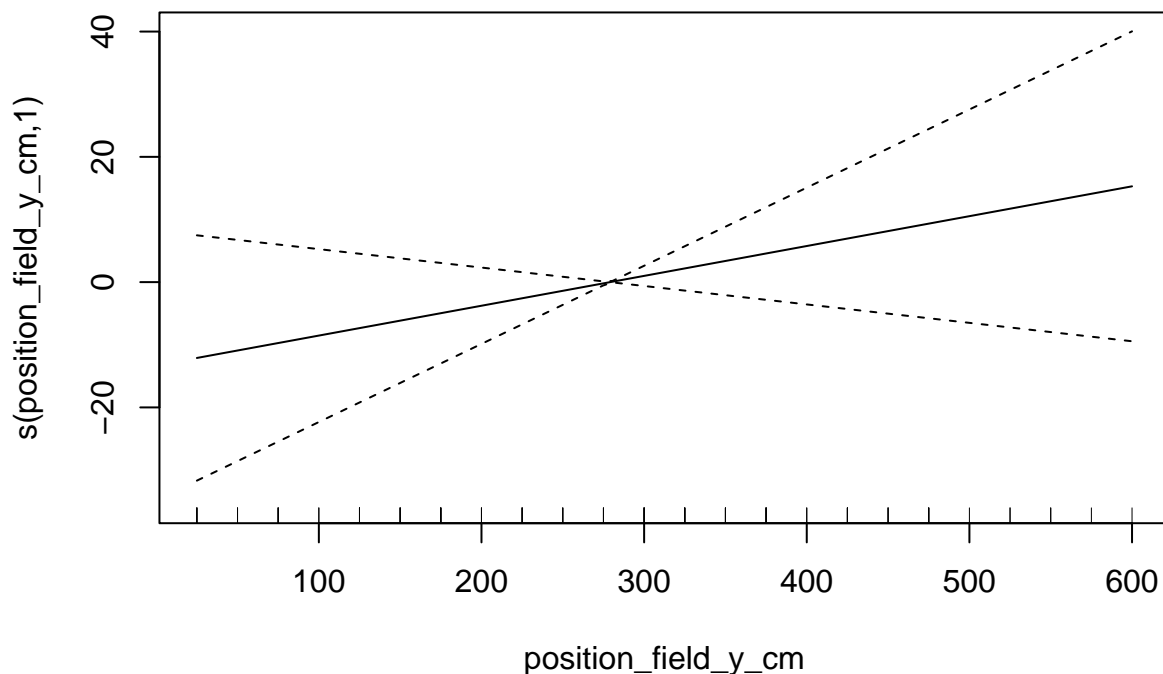
GCV = 2607.6 Scale est. = 2346.8 n = 80

```

plot(gam.cob_weight.num.add)

```





The summary reveals that the estimated coefficients are identical to those in the previous model, emphasizing that the previous model simply added the two smooth terms in an additive manner.

We now consider the variables *seed\_coord\_x* and *seed\_coord\_y* in a linear manner.

```
lm.cob_weight.num <- lm(cob_weight.num ~ position_field_x_cm +
                        position_field_y_cm +
                        soil.fac +
                        depth +
                        seed.weight,
                        data = d.maize)

##
summary(lm.cob_weight.num)
```

Call:

```
lm(formula = cob_weight.num ~ position_field_x_cm + position_field_y_cm +
    soil.fac + depth + seed.weight, data = d.maize)
```

Residuals:

Min	1Q	Median	3Q	Max
-88.5649	-25.2391	3.5335	22.7456	197.7709

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	131.569262	37.084313	3.5478	0.0006881 ***
position_field_x_cm	-0.153161	0.086483	-1.7710	0.0807941 .

```

position_field_y_cm    0.047660    0.038541    1.2366 0.2202611
soil.facComposana      31.419137    17.628538    1.7823 0.0789182 .
soil.facherbs          21.649510    14.480075    1.4951 0.1392524
soil.facmixture         7.408432    16.770561    0.4418 0.6599925
depth                  0.381387     3.328376    0.1146 0.9090919
seed.weight            -0.809831     0.939307   -0.8622 0.3914628

```

```

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 48.444 on 72 degrees of freedom

(28 observations deleted due to missingness)

Multiple R-squared: 0.10995, Adjusted R-squared: 0.023414

F-statistic: 1.2706 on 7 and 72 DF, p-value: 0.27711

The estimated coefficients remain unchanged, confirming the hypothesis that the original model was simply introducing the smooth terms in a linear manner.

To further confirm this conclusion, we will compare the three models using the AIC and BIC criteria. We use this method instead of comparing the models using the `anova()` method because the models are not nested. Indeed, when fitting a GAM with smooth terms, if we modify the these smooth terms, the smooths change, and as a result, the models are not nested.

```

AIC(gam.cob_weight.num.add,
    gam.cob_weight.num,
    lm.cob_weight.num)

```

```

              df      AIC
gam.cob_weight.num.add  9 857.46642
gam.cob_weight.num      9 857.46642
lm.cob_weight.num       9 857.46642

```

```

##
BIC(gam.cob_weight.num.add,
    gam.cob_weight.num,
    lm.cob_weight.num)

```

```

              df      BIC
gam.cob_weight.num.add  9 878.90466
gam.cob_weight.num      9 878.90466
lm.cob_weight.num       9 878.90466

```

We keep the simpler model.

The next step is to verify whether the variable *soil.fac* has an influence on the response variable as a whole. Indeed, the above summary only shows the relative influence of each level compared to the reference level of *soil.fac*.

To achieve this result, we use the `drop1()` function.

```

drop1(lm.cob_weight.num, test = "F")

```

Single term deletions

Model:

```

cob_weight.num ~ position_field_x_cm + position_field_y_cm +
    soil.fac + depth + seed.weight
              Df Sum of Sq  RSS      AIC F value    Pr(>F)
<none>                168971 628.436

```

```

position_field_x_cm 1 7360.58 176331 629.847 3.13642 0.080794 .
position_field_y_cm 1 3588.59 172559 628.118 1.52913 0.220261
soil.fac            3 9840.91 178812 626.965 1.39777 0.250477
depth              1 30.81 169001 626.451 0.01313 0.909092
seed.weight        1 1744.43 170715 627.258 0.74332 0.391463

```

---

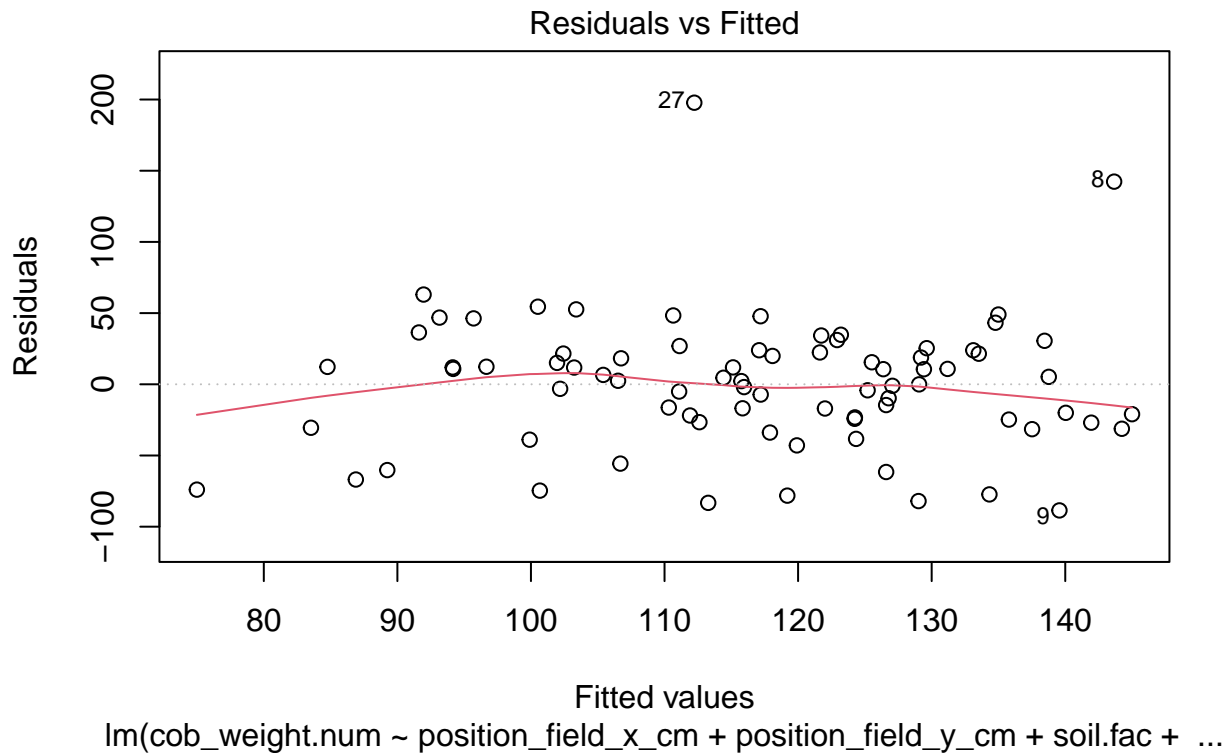
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

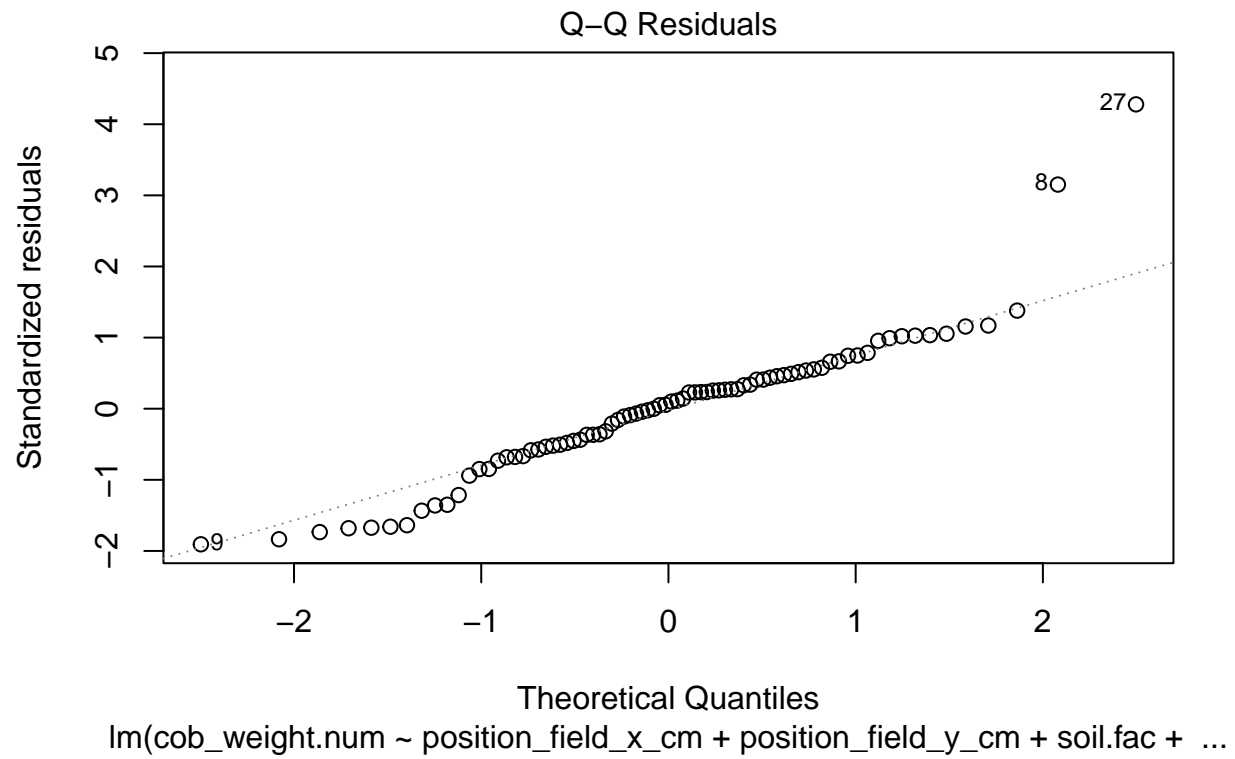
The variable *soil.fac* does not show evidence of having a significant influence on the response variable.

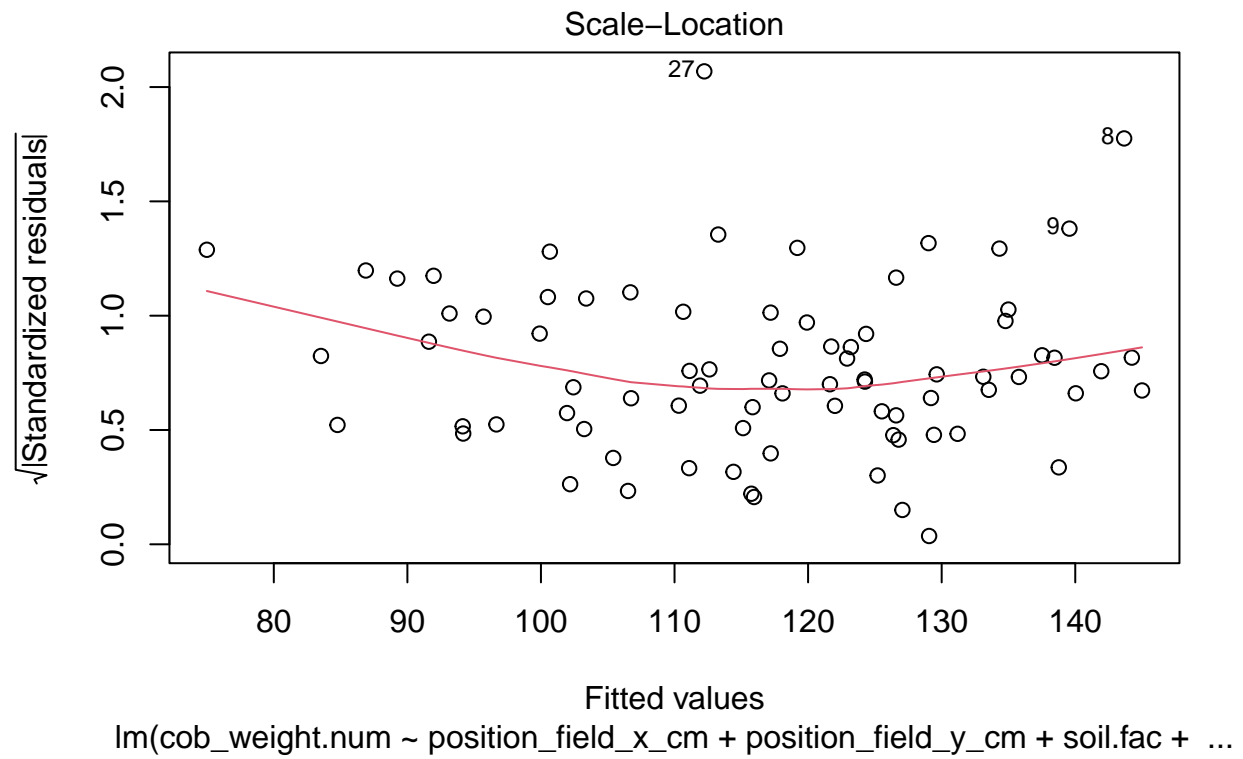
However, since the variable is in the design, we do not drop it from the model.

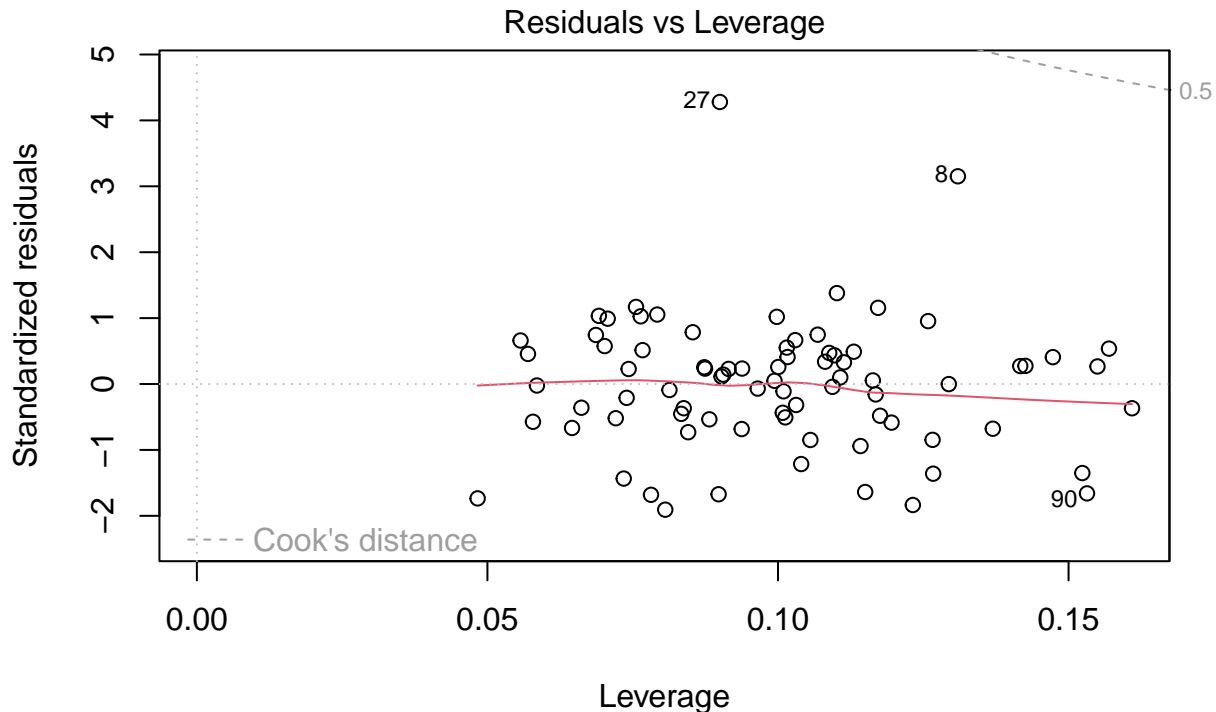
## 6.4 Model checking

```
plot(lm.cob_weight.num)
```









The residuals plot do not show evidence of a departure from the model assumptions.

#### 6.4.1 Confidence intervals

```
( CI.cob_weight <- confint(lm.cob_weight.num) )
```

	2.5 %	97.5 %
(Intercept)	57.643036412	205.495488408
position_field_x_cm	-0.325562740	0.019240015
position_field_y_cm	-0.029171294	0.124490404
soil.facComposana	-3.722711085	66.560985855
soil.facherbs	-7.215993195	50.515012613
soil.facmixture	-26.023069917	40.839933364
depth	-6.253609884	7.016383083
seed.weight	-2.682305360	1.062642626

Firstly, we create a data frame that includes the parameter estimates and their corresponding confidence intervals.

```
## Store the estimated values as dataframe
( d.coef.cob_weight <- data.frame(coef.cob_weight = coef(lm.cob_weight.num)) )
```

	coef.cob_weight
(Intercept)	131.569262410
position_field_x_cm	-0.153161362
position_field_y_cm	0.047659555
soil.facComposana	31.419137385
soil.facherbs	21.649509709

```

soil.facmixture      7.408431723
depth                0.381386600
seed.weight          -0.809831367

##
d.CI.cob_weight <- as.data.frame(CI.cob_weight)

## Join the two dataframe by rowname
d.est.cob_weight <- left_join(rownames_to_column(d.coef.cob_weight),
                             rownames_to_column(d.CI.cob_weight),
                             by = c("rowname" = "rowname"))

##
## visualise the dataframe
d.est.cob_weight %>%
  kable(caption = paste0("Estimates and 95\\% CI."),
        label = "tab_coef_cob_weight",
        booktabs = TRUE,
        longtable = TRUE,
        linesep = c("")) %>%
  # landscape() %>%
  kable_styling(font_size = 7,
                latex_options = c("striped", "repeat_header", "hold_position"))

```

Table 1: Estimates and 95% CI.

rowname	coef.cob_weight	2.5 %	97.5 %
(Intercept)	131.56926241	57.64303641	205.49548841
position_field_x_cm	-0.15316136	-0.32556274	0.01924002
position_field_y_cm	0.04765956	-0.02917129	0.12449040
soil.facComposana	31.41913739	-3.72271109	66.56098586
soil.facherbs	21.64950971	-7.21599320	50.51501261
soil.facmixture	7.40843172	-26.02306992	40.83993336
depth	0.38138660	-6.25360988	7.01638308
seed.weight	-0.80983137	-2.68230536	1.06264263

We provide an example on how to read this table.

An increase of 1 cm in the variable *depth* is associated with a 0.3813866 grams increase in cob weight, by keeping all other variables constant.

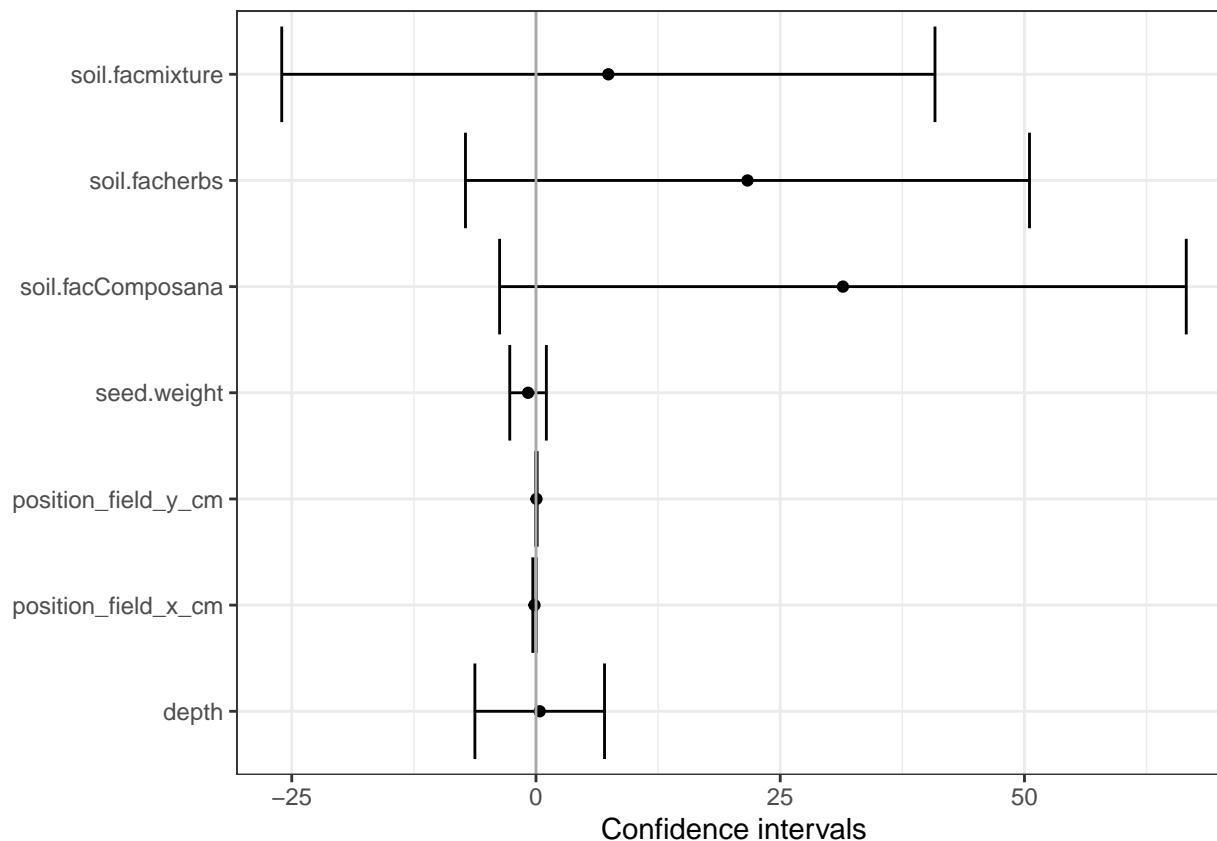
We visualise the estimated values of the model along with their corresponding confidence intervals. This will provide us with a better understanding of which variables have a more significant influence on the variable *cob\_weight*.

We do not plot the intercept because it lacks practical interpretation. The intercept represents the reference level, corresponding, between other features, to a cob with seed weight of 0 gr., which is impossible.

```

d.est.cob_weight %>%
  filter(rowname != "(Intercept)") %>%
  ggplot(mapping = aes(y = rowname, x = coef.cob_weight)) +
  geom_point() +
  geom_errorbar(mapping = aes(xmin = `2.5 %`, xmax = `97.5 %`)) +
  xlab("Confidence intervals") +
  theme(axis.title.y = element_blank()) +
  geom_vline(xintercept = 0, color = "darkgrey")

```



## 6.5 Contrasts

We now want to further investigate the soil effect.

We want to compare all the soils in a pairwise manner.

With this purpose in mind, we use the `glht()` function from the `{multcomp}` package.

```
test.soil <- glht(lm.cob_weight.num,
                 linfct = mcp(soil.fac = "Tukey"))
```

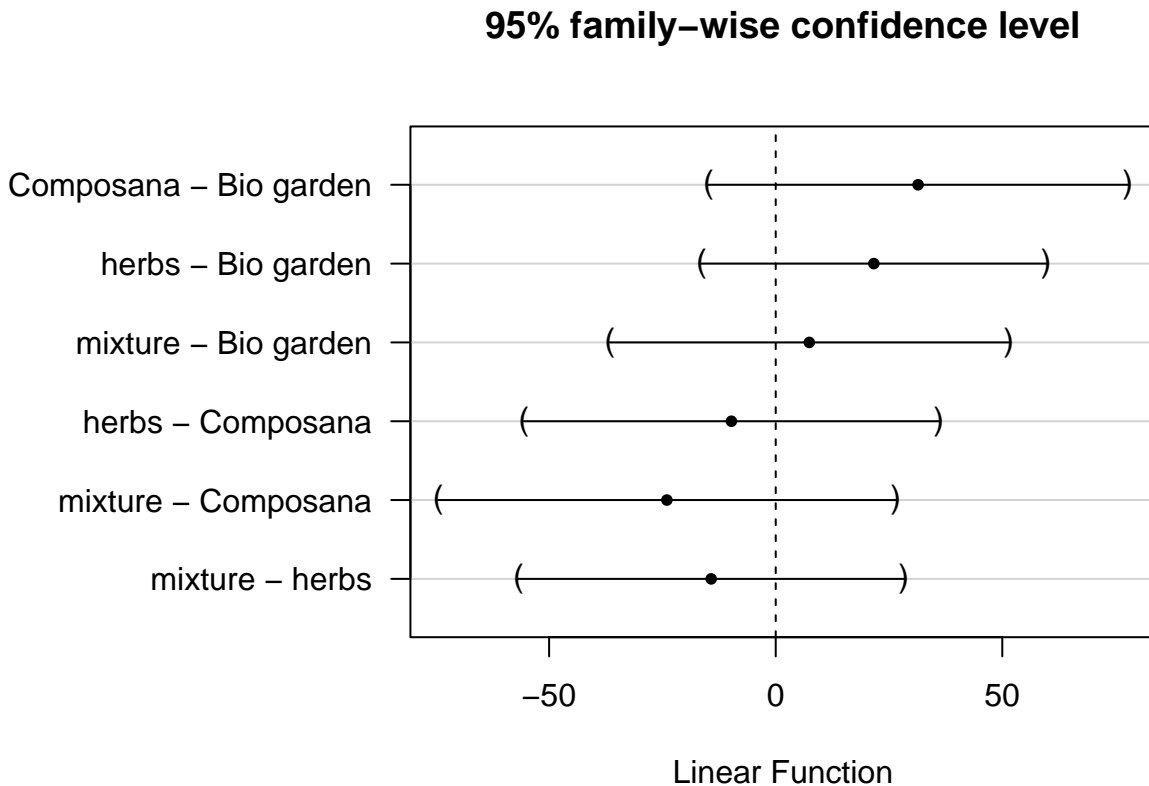
These tests can also be visualised in a graphical manner.

Note that the margins of the plotting area are adjusted to make all test names fit in it.

```
par("mar") ## the second value refers to the left margin (to be enlarged)
```

```
[1] 5.1 4.1 4.1 2.1
```

```
par(mar = c(5.1, 11, 4.1, 2.1))
plot(test.soil)
```



There is no evidence indicating that the types of soil are statistically different at the 5% significance level.

## 7 Methods description

To comprehend the factors that influence cob weight, we initially employed a Generalized Additive Model (GAM). However, after conducting a thorough analysis, it was simplified into a linear model.

We started with an additive approach because the explanatory variables *position\_field\_x\_cm* and *position\_field\_y\_cm* did not follow any clear distribution, thus they were introduced in the model as smooth terms.

The statistical analysis was performed using the R programming language, specifically version 4.3.1 (see citation below). The generalised additive model was fitted with the `gam()` function in the `{mgcv}` add-on package (see citation below).

### Citations

`citation()`

To cite R in publications use:

R Core Team (2023). *\_R\_: A Language and Environment for Statistical Computing\_*. R Foundation for Statistical Computing, Vienna, Austria.  
<<https://www.R-project.org/>>.

A BibTeX entry for LaTeX users is

```
@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2023},
  url = {https://www.R-project.org/},
}
```

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also 'citation("pkgname")' for citing R packages.

```
citation("mgcv")
```

2011 for generalized additive model method; 2016 for beyond exponential family; 2004 for strictly additive GCV based model method and basics of gamm; 2017 for overview; 2003 for thin plate regression splines.

Wood, S.N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)* 73(1):3-36

Wood S.N., N. Pya and B. Saefken (2016) Smoothing parameter and model selection for general smooth models (with discussion). *Journal of the American Statistical Association* 111:1548-1575.

Wood, S.N. (2004) Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*. 99:673-686.

Wood, S.N. (2017) *Generalized Additive Models: An Introduction with R* (2nd edition). Chapman and Hall/CRC.

Wood, S.N. (2003) Thin-plate regression splines. *Journal of the Royal Statistical Society (B)* 65(1):95-114.

To see these entries in BibTeX format, use 'print(<citation>, bibtex=TRUE)', 'toBibtex(.)', or set 'options(citation.bibtex.max=999)'.

## 8 Session information

```
sessionInfo()
```

```
R version 4.3.1 (2023-06-16)
```

```
Platform: aarch64-apple-darwin20 (64-bit)
```

```
Running under: macOS Sonoma 14.0
```

```
Matrix products: default
```

```
BLAS: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
```

```
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib; LAPACK vers
```

```
locale:
```

```
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

```
time zone: Europe/Zurich
```

```
tzcode source: internal
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] mgcv_1.8-42      nlme_3.1-162      multcomp_1.4-25  TH.data_1.1-2  
[5] MASS_7.3-60      survival_3.5-5    mvtnorm_1.2-3    tibble_3.2.1  
[9] ggplot2_3.4.4    kableExtra_1.3.4 dplyr_1.1.3      knitr_1.44
```

```
loaded via a namespace (and not attached):
```

```
[1] sandwich_3.0-2    utf8_1.2.4        generics_0.1.3    xml2_1.3.5  
[5] stringi_1.7.12    lattice_0.21-8    digest_0.6.33     magrittr_2.0.3  
[9] evaluate_0.22     grid_4.3.1        fastmap_1.1.1     Matrix_1.6-1.1  
[13] httr_1.4.7        rvest_1.0.3       fansi_1.0.5       viridisLite_0.4.2  
[17] scales_1.2.1      codetools_0.2-19  cli_3.6.1         rlang_1.1.1  
[21] munsell_0.5.0     splines_4.3.1     withr_2.5.1       yaml_2.3.7  
[25] tools_4.3.1       colorspace_2.1-0  webshot_0.5.5     vctrs_0.6.4  
[29] R6_2.5.1          zoo_1.8-12        lifecycle_1.0.3   stringr_1.5.0  
[33] pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.4      glue_1.6.2  
[37] systemfonts_1.0.5 xfun_0.40          tidyselect_1.2.0  rstudioapi_0.15.0  
[41] farver_2.1.1      htmltools_0.5.6.1 rmarkdown_2.25    svglite_2.1.2  
[45] labeling_0.4.3    compiler_4.3.1
```