

# Modelling Maize Project: *germinated.in.lab*

Modelling binary response

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>germinated.in.lab</i></b>	<b>5</b>
6.1	Aim . . . . .	5
6.2	Model fitting . . . . .	6
6.3	Model checking . . . . .	7
6.3.1	Random effects . . . . .	7
6.3.2	Fixed effects . . . . .	10
6.4	Contrasts . . . . .	13
<b>7</b>	<b>Methods description</b>	<b>14</b>
<b>8</b>	<b>Session information</b>	<b>16</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) ## function glht
library(lme4) ## function glmer
```

# 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: *germinated.in.lab*

### 6.1 Aim

We are interested in testing whether *germinated.in.lab* is influenced by the following variables:

- Soil (variable *soil.fac*)
- Depth in soil (variable *depth*)
- Seed weight (variable *seed.weight*)

While controlling for

- pot (variable *pot.fac*)
- well (variable *well.fac*)

Which are considered as random effects.

## 6.2 Model fitting

The variable *germinated.in.lab* is a binary variable, for this reason we fit a binomial model.

Based on the graphical analysis, it is not necessary to transform any of the explanatory variables. Therefore, they can be used as they are in the model.

*depth* and *seed.weight* are numeric variables which are introduced in the model as continuous variables.

The variable representing the type of soil has 4 levels, and it is thus introduced as a factor in the model.

Moreover, we introduce *pot.fac* and *well.fac* as random effects.

```
bin.germinated.in.lab <- glmer(germinated.in.lab ~
                                soil.fac +
                                depth +
                                seed.weight +
                                (1 | pot.fac) +
                                (1 | well.fac),
                                family = "binomial",
                                data = d.maize)
```

boundary (singular) fit: see help('isSingular')

```
##
```

```
summary(bin.germinated.in.lab)
```

Generalized linear mixed model fit by maximum likelihood (Laplace  
Approximation) [glmerMod]

Family: binomial ( logit )

Formula: germinated.in.lab ~ soil.fac + depth + seed.weight + (1 | pot.fac) +  
(1 | well.fac)

Data: d.maize

AIC	BIC	logLik	deviance	df.resid
127.9	149.4	-56.0	111.9	100

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.45803	0.35947	0.45780	0.53973	0.80587

Random effects:

Groups	Name	Variance	Std.Dev.
pot.fac	(Intercept)	1.3014e-09	3.6074e-05
well.fac	(Intercept)	2.2141e-01	4.7055e-01

Number of obs: 108, groups: pot.fac, 18; well.fac, 6

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	2.323508	1.486111	1.5635	0.1179
soil.facComposana	-0.523136	0.733782	-0.7129	0.4759
soil.facherbs	-0.213793	0.701361	-0.3048	0.7605
soil.facmixture	-0.760716	0.720600	-1.0557	0.2911
depth	-0.048714	0.140972	-0.3456	0.7297
seed.weight	-0.015669	0.041846	-0.3744	0.7081

Correlation of Fixed Effects:

(Intr) sl.fcC sl.fch sl.fcm depth

```

sol.fcCmpsn -0.286
soil.fchrbs -0.365  0.597
soil.fcmxtr -0.337  0.581  0.611
depth      -0.337 -0.027 -0.009  0.003
seed.weight -0.853  0.015  0.087  0.055 -0.001
optimizer (Nelder_Mead) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')

```

The variance of the random effects is really small; for this reason we have a warning message. Nonetheless, this variance is not numerically equal to zero, thus we keep the random effects in the model.

No estimate shows evidence of being significantly different from zero.

## 6.3 Model checking

### 6.3.1 Random effects

We start by extrapolating all the random effects of the model.

```
(ranef.germinated.in.lab <- ranef(bin.germinated.in.lab))
```

```

$pot.fac
  (Intercept)
A1  1.3759659e-09
A2 -1.1938374e-09
A3  2.1446874e-10
A4  1.2700112e-09
A5  2.7511764e-10
A6  1.5500236e-10
B1 -2.9789458e-10
B2  8.7608965e-10
B3 -6.1688450e-10
B4 -6.5884974e-10
B5  1.4189500e-10
B6  9.3605524e-10
C1 -1.5980968e-09
C2 -1.3624358e-09
C3 -1.4089434e-09
C4 -7.4169388e-10
C5  1.9119352e-09
C6  2.2053000e-10

```

```

$well.fac
  (Intercept)
a -0.387844042
b  0.122260767
c  0.093378996
d -0.413461068
e  0.093129025
f  0.407199828

```

with conditional variances for "pot.fac" "well.fac"

We extract the variance of these random effects.

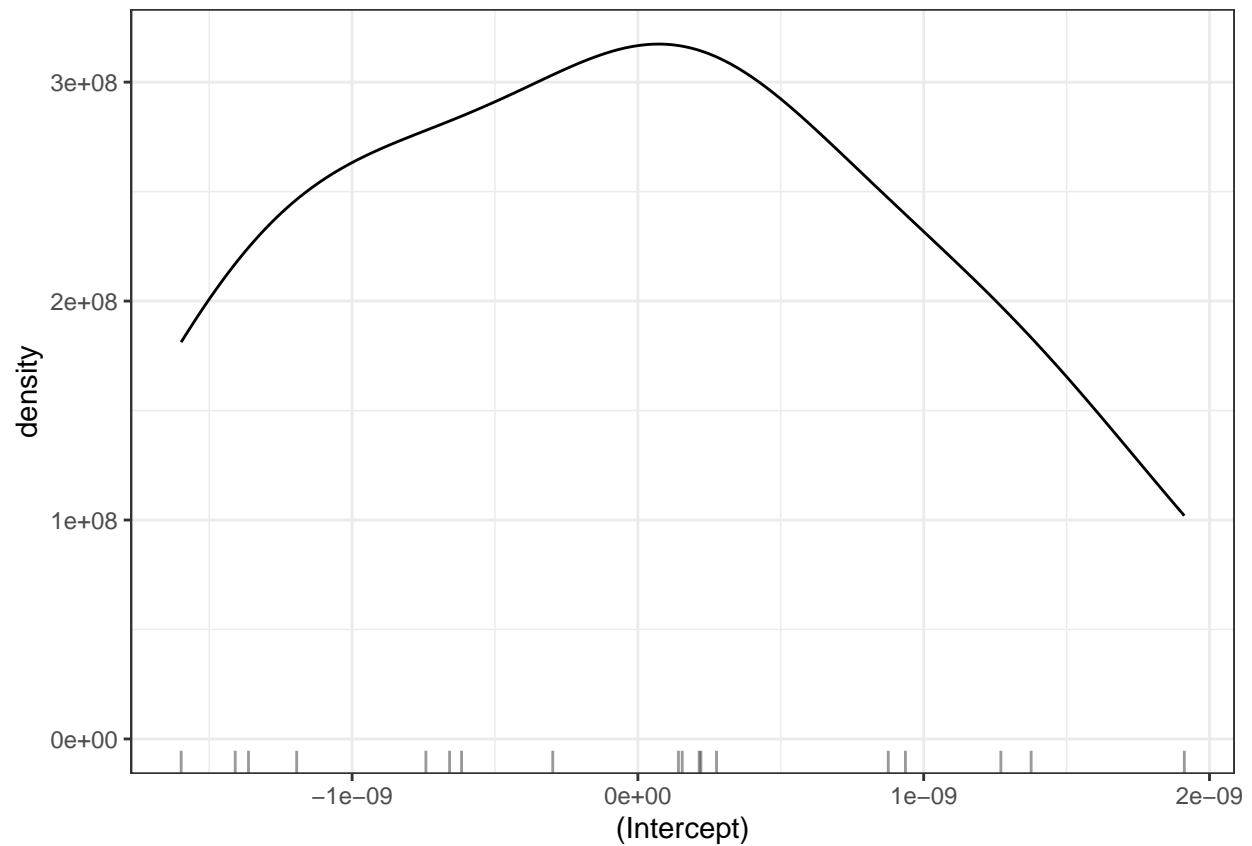
```
(st.ranef <- VarCorr(bin.germinated.in.lab))
```

Groups	Name	Std.Dev.
--------	------	----------

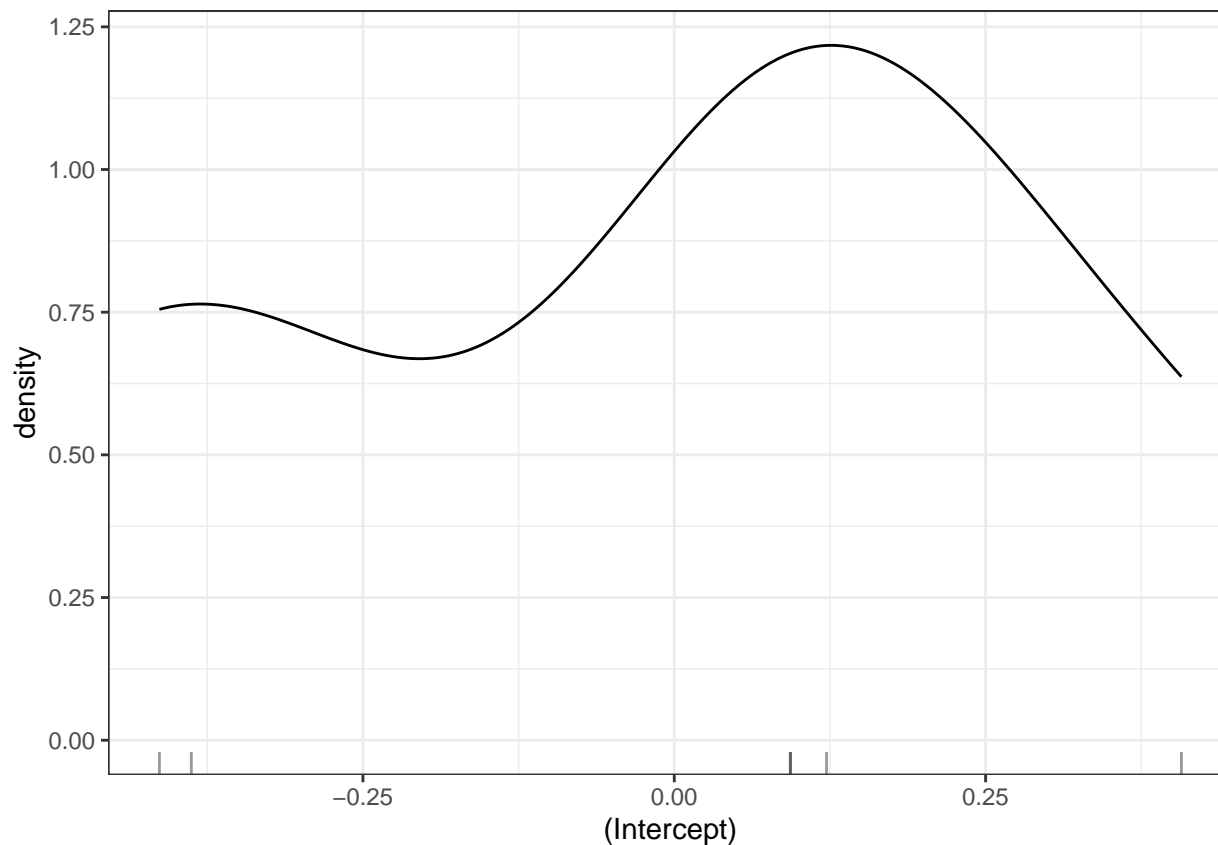
```
pot.fac (Intercept) 3.60743e-05  
well.fac (Intercept) 4.70545e-01
```

We also plot their densities, to see how they are distributed.

```
ggplot(ranef.germinated.in.lab$pot.fac, mapping = aes(x = `(Intercept)`)) +  
  geom_density() +  
  geom_rug(alpha = 0.4)
```



```
##  
ggplot(ranef.germinated.in.lab$well.fac, mapping = aes(x = `(Intercept)`)) +  
  geom_density() +  
  geom_rug(alpha = 0.4)
```



We look at the range of values they take.

```
range(ranef.germinated.in.lab$pot.fac)
```

```
[1] -1.5980968e-09  1.9119352e-09
```

```
range(ranef.germinated.in.lab$well.fac)
```

```
[1] -0.41346107  0.40719983
```

At last, we extract their confidence intervals.

```
( CI.ranef.germinated.in.lab <- confint(bin.germinated.in.lab, parm = "theta_",  
                                         method = "profile") )
```

Computing profile confidence intervals ...

Error in zeta(shiftpar, start = opt[seqpar1][-w]): profiling detected new, lower deviance (deviance dif.

```
## error
```

```
( CI.ranef.germinated.in.lab <- confint(bin.germinated.in.lab, parm = "theta_",  
                                         method = "Wald") )
```

```
      2.5 % 97.5 %  
.sig01    NA     NA  
.sig02    NA     NA
```

```
## Only NAs
```

```
set.seed(2023)
```

```
( CI.ranef.germinated.in.lab <- confint(bin.germinated.in.lab, parm = "theta_",  
                                         method = "boot") )
```

```
Computing bootstrap confidence intervals ...
```

```
Warning in bootMer(object, FUN = FUN, nsim = nsim, ...): some bootstrap runs  
failed (5/500)
```

```
300 message(s): boundary (singular) fit: see help('isSingular')
```

```
150 warning(s): Model failed to converge with max|grad| = 0.00200132 (tol = 0.002, component 1) (and others)
```

```
5 error(s): Datedated VtV is not positive definite
```

```
          2.5 %      97.5 %  
.sig01      0 0.67676124  
.sig02      0 1.08891704
```

Calculating profile and Wald confidence intervals does not work; indeed, the first one returns an error, and the second one only NAs. This is the case because the variance of the random effects is really small, consequently we are too close to the boundary for the algorithms to correctly converge.

We achieve calculating bootstrap confidence intervals, but we can observe that we have a lot of warnings for this method too.

### 6.3.2 Fixed effects

We now examine whether dropping any single fixed-effect variable would have an influence on the model. With this purpose in mind, we also perform an asymptotic likelihood ratio test.

```
drop1(bin.germinated.in.lab, test = "Chisq")
```

```
boundary (singular) fit: see help('isSingular')  
boundary (singular) fit: see help('isSingular')  
boundary (singular) fit: see help('isSingular')
```

```
Single term deletions
```

```
Model:
```

```
germinated.in.lab ~ soil.fac + depth + seed.weight + (1 | pot.fac) +  
  (1 | well.fac)
```

	npar	AIC	LRT	Pr(Chi)
<none>		127.915		
soil.fac	3	123.311	1.395856	0.70651
depth	1	126.034	0.118977	0.73015
seed.weight	1	126.054	0.138691	0.70959

None of the fixed-effect variables show evidence of having an influence on the model, when considering the others present in the model.

#### 6.3.2.1 Confidence intervals

We compute the confidence intervals for the fixed effect estimates.

```
( CI.germinated.in.lab <- confint(bin.germinated.in.lab, parm = "beta_") )
```

```
Computing profile confidence intervals ...
```

	2.5 %	97.5 %
(Intercept)	-0.522231007	5.410019021
soil.facComposana	-2.044562520	0.908695882
soil.facherbs	-1.682956994	1.142104904
soil.facmixture	-2.267093280	0.627582842
depth	-0.331827234	0.228578277
seed.weight	-0.098736501	0.068103772

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

```
## Store the estimated values as dataframe
( d.coef.germinated.in.lab <-
  data.frame(fixef.germinated.in.lab = fixef(bin.germinated.in.lab)) )

                fixef.germinated.in.lab
(Intercept)          2.323507825
soil.facComposana    -0.523135942
soil.facherbs        -0.213793070
soil.facmixture      -0.760716298
depth                -0.048713700
seed.weight          -0.015669006

##
d.CI.germinated.in.lab <- as.data.frame(CI.germinated.in.lab)

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

##
## visualise the dataframe
d.est.germinated.in.lab %>%
  kable(caption = paste0("Estimates and 95\\% CI."),
        label = "tab_coef_germinated.in.lab",
        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	fixef.germinated.in.lab	2.5 %	97.5 %
(Intercept)	2.32350782	-0.52223101	5.41001902
soil.facComposana	-0.52313594	-2.04456252	0.90869588
soil.facherbs	-0.21379307	-1.68295699	1.14210490
soil.facmixture	-0.76071630	-2.26709328	0.62758284
depth	-0.04871370	-0.33182723	0.22857828
seed.weight	-0.01566901	-0.09873650	0.06810377

These estimates are in the linear predictor scale, making it difficult to interpret.

To address this, we apply the inverse of the link function to obtain a multiplicative effect of the coefficients. This transformation allows for easier interpretability of the results.

```
## exponential
d.est.germinated.in.lab.exp <- data.frame(d.est.germinated.in.lab$rowname,
                                           exp(d.est.germinated.in.lab$fixef.germinated.in.lab),
                                           exp(d.est.germinated.in.lab$`2.5 %`),
                                           exp(d.est.germinated.in.lab$`97.5 %`))

##
colnames(d.est.germinated.in.lab.exp) <- c("rowname",
                                           "exp.fixef.germinated.in.lab",
```

```

                                "2.5 %",
                                "97.5 %")
d.est.germinated.in.lab.exp %>%
  kable(caption = paste0("Exponential of the estimates and 95\\% CI."),
        label = "tab_exp_fixef_germinated_in_lab",
        booktabs = TRUE,
        longtable = TRUE,
        linesep = c(" ")) %>%
  # landscape() %>%
  kable_styling(font_size = 7,
                latex_options = c("striped", "repeat_header", "hold_position"))

```

Table 2: Exponential of the estimates and 95% CI.

rowname	exp.fixef.germinated.in.lab	2.5 %	97.5 %
(Intercept)	10.21143147	0.59319565	223.6358414
soil.facComposana	0.59265909	0.12943680	2.4810848
soil.facherbs	0.80751547	0.18582368	3.1333568
soil.facmixture	0.46733156	0.10361292	1.8730776
depth	0.95245378	0.71761129	1.2568119
seed.weight	0.98445311	0.90598140	1.0704764

Let make an example to better understand this table.

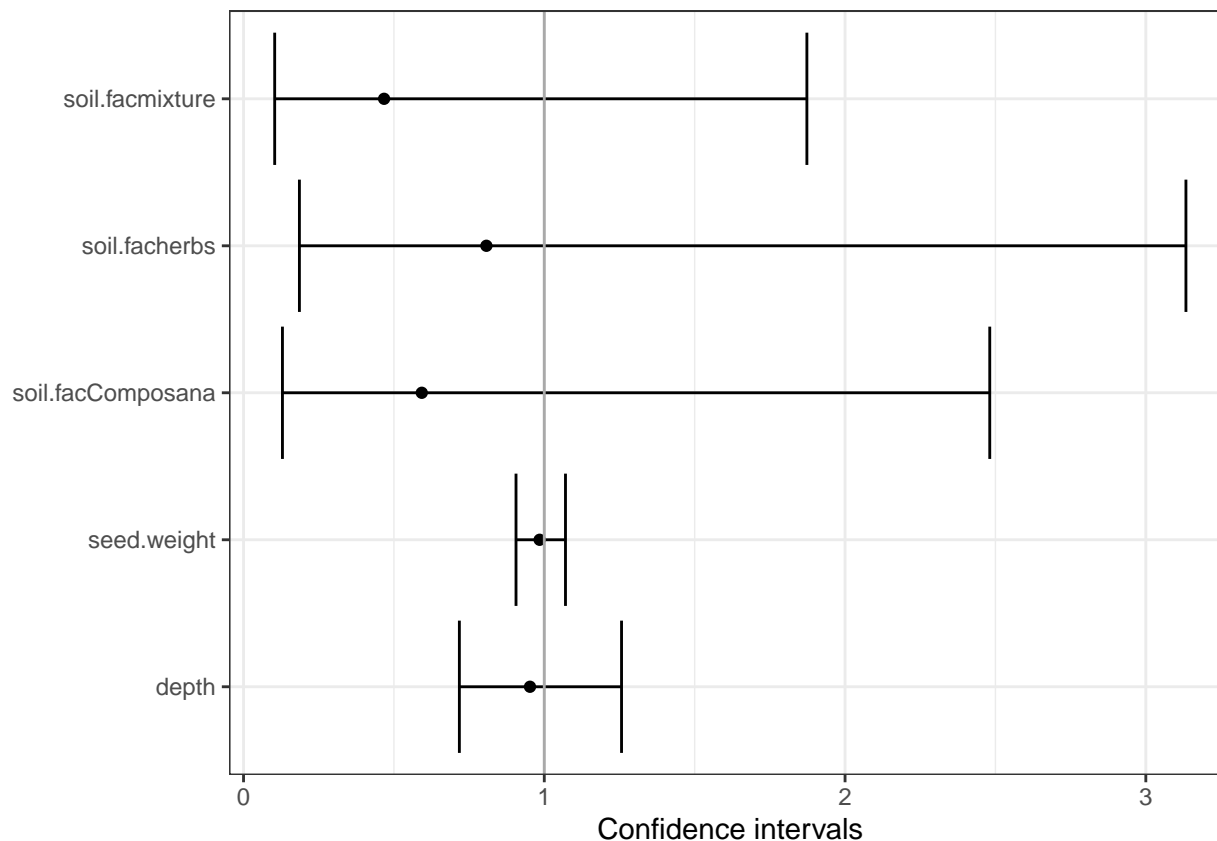
Taking the variable *depth*, it is possible to observe that its multiplicative effect is 95.2 %, and the confidence interval goes from -28.2 % to 26 %.

This means that increasing the seed depth by one centimeter will reduce the odds of a seed germinating in the lab by about 5%. We can visualise this in a table by plotting it.

```

## Plot
d.est.germinated.in.lab.exp %>%
  filter( rowname != "(Intercept)" ) %>%
  ggplot(mapping = aes(y = rowname, x = exp.fixef.germinated.in.lab)) +
  geom_point() +
  geom_errorbar(mapping = aes(xmin = `2.5 %`, xmax = `97.5 %`)) +
  xlab("Confidence intervals") +
  theme(axis.title.y = element_blank()) +
  geom_vline(xintercept = 1, color = "darkgrey")

```



Here the vertical line is set at 1, which corresponds to no effect.

## 6.4 Contrasts

We now want to further investigate the soil effect in a pairwise manner.

```
test.soil <- multcomp::glht(bin.germinated.in.lab,
                             linfct = mcp(soil.fac = "Tukey"))
##
summary(test.soil)
```

### Simultaneous Tests for General Linear Hypotheses

#### Multiple Comparisons of Means: Tukey Contrasts

```
Fit: glmer(formula = germinated.in.lab ~ soil.fac + depth + seed.weight +
            (1 | pot.fac) + (1 | well.fac), data = d.maize, family = "binomial")
```

#### Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z )
Composana - Bio garden == 0	-0.52314	0.73378	-0.7129	0.8916
herbs - Bio garden == 0	-0.21379	0.70136	-0.3048	0.9901
mixture - Bio garden == 0	-0.76072	0.72060	-1.0557	0.7157
herbs - Composana == 0	0.30934	0.64503	0.4796	0.9635
mixture - Composana == 0	-0.23758	0.66598	-0.3567	0.9844

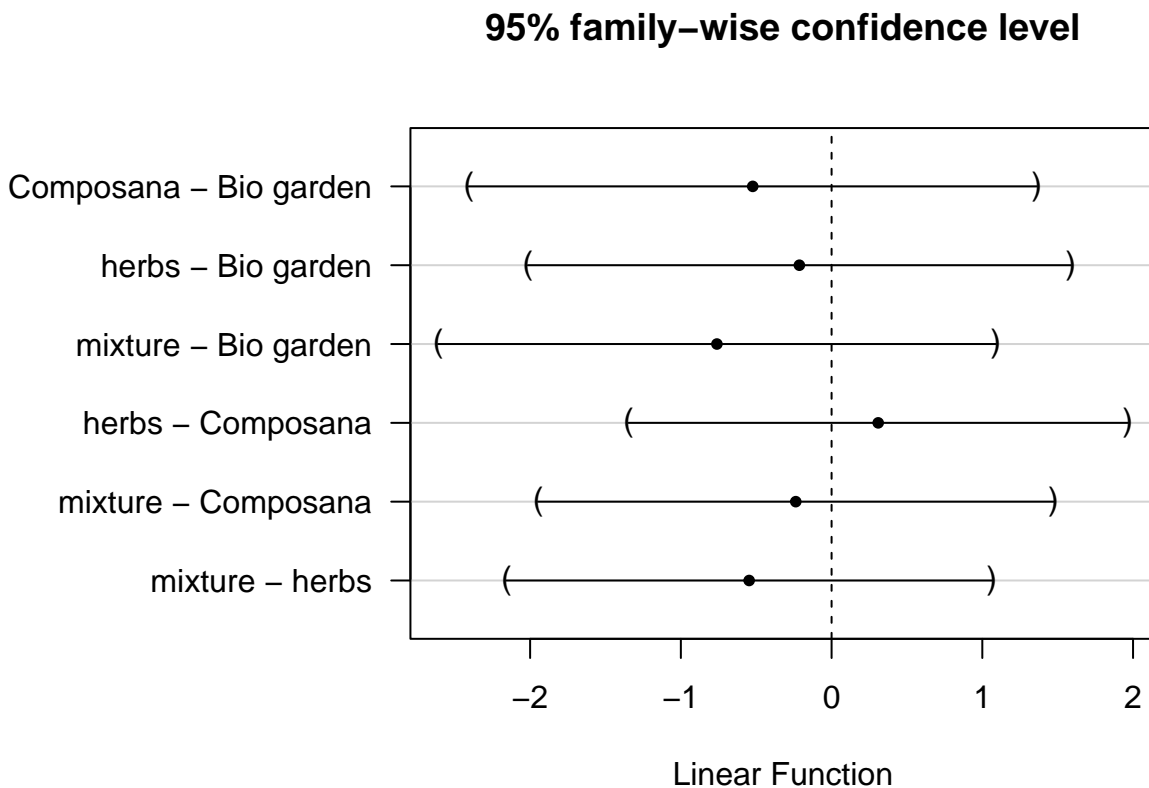
```
mixture - herbs == 0      -0.54692    0.62722 -0.8720    0.8188
(Adjusted p values reported -- single-step method)
```

This test can also be visualised in a plot.

```
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)
```



We fail to reject the null hypothesis that there is no statistically significant difference among the types of soil at the 5% significance level.

This result is consistent with the findings from the `summary()` and `drop1()` functions, which do not provide evidence of a difference between the soil types.

## 7 Methods description

To understand which factors influence the germination time of the seeds, a generalised mixed-effects model was employed. This approach was used because the response variable, *germinated.in.lab*, is binary.

To account for the seeds clustered in pots, and in wells, a mixed model approach was adopted (i.e. *pot.fac* and *well.fac* were taken as random effects).

The statistical analysis was performed using the R programming language, specifically version 4.3.1 (see

citation below). The generalised mixed-effects model was fitted with the `glmer()` function in the `{lme4}` 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("lme4")
```

To cite lme4 in publications use:

```
Douglas Bates, Martin Maechler, Ben Bolker, Steve Walker (2015).
Fitting Linear Mixed-Effects Models Using lme4. Journal of
Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01.
```

A BibTeX entry for LaTeX users is

```
@Article{,
  title = {Fitting Linear Mixed-Effects Models Using {lme4}},
  author = {Douglas Bates and Martin M{"a}chler and Ben Bolker and Steve Walker},
  journal = {Journal of Statistical Software},
  year = {2015},
  volume = {67},
  number = {1},
  pages = {1--48},
  doi = {10.18637/jss.v067.i01},
}
```

## 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] lme4_1.1-34      Matrix_1.6-1.1  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   httr_1.4.7  
[13] rvest_1.0.3       fansi_1.0.5     viridisLite_0.4.2 scales_1.2.1  
[17] codetools_0.2-19 cli_3.6.1        rlang_1.1.1     munsell_0.5.0  
[21] splines_4.3.1     withr_2.5.1     yaml_2.3.7      tools_4.3.1  
[25] nloptr_2.0.3      minqa_1.2.6     colorspace_2.1-0 webshot_0.5.5  
[29] boot_1.3-28.1     vctr_0.6.4      R6_2.5.1        zoo_1.8-12  
[33] lifecycle_1.0.3   stringr_1.5.0   pkgconfig_2.0.3 pillar_1.9.0  
[37] gtable_0.3.4      Rcpp_1.0.11     glue_1.6.2      systemfonts_1.0.5  
[41] xfun_0.40         tidyselect_1.2.0 rstudioapi_0.15.0 farver_2.1.1  
[45] htmltools_0.5.6.1 nlme_3.1-162    labeling_0.4.3  rmarkdown_2.25  
[49] svglite_2.1.2     compiler_4.3.1
```