

Graphical Analysis: Maize Project

Authors: Dr. Luisa Barbanti & Nisia Trisconi | Zurich Data Scientists

Reviewer: Dr. Matteo Tanadini | Zurich Data Scientists

October 30, 2023

Contents

1	Freezing Package versions	3
2	Load packages	3
3	Settings	3
4	Getting data	4
5	Table 1	5
5.1	Categorical variables	5
5.2	Continuous variables	7
6	Graphical Analysis	9
6.1	Response variable: <i>cob_weight.num</i>	12
6.1.1	We look at <i>pot.fac</i>	12
6.1.2	We look at <i>soil.fac</i>	12
6.1.3	We look at <i>well.fac</i>	13
6.1.4	We look at <i>depth</i>	14
6.1.5	We look at <i>seed.weight.grams</i>	15
6.1.6	We look at <i>broken</i>	16
6.1.7	We look at <i>date.germinated.asDate</i>	17
6.1.8	We look at <i>germinated.in.lab</i>	18
6.1.9	We look at <i>germinated.in.field</i>	19
6.1.10	We look at the coordinates	20
6.1.11	We look at the position in the field	22
6.2	Response variable: <i>height_2022_07_05.num</i>	24
6.2.1	We look at <i>pot.fac</i>	24
6.2.2	We look at <i>soil.fac</i>	25
6.2.3	We look at <i>well.fac</i>	26
6.2.4	We look at <i>depth</i>	27
6.2.5	We look at <i>seed.weight.grams</i>	28
6.2.6	We look at <i>broken</i>	29
6.2.7	We look at <i>germinated.in.lab</i>	30
6.2.8	We look at <i>germinated.in.field</i>	31
6.2.9	We look at the coordinates	32
6.3	We look at the position in the field	34
6.4	Response variable: <i>germinated.yes</i>	36
6.4.1	We look at <i>pot.fac</i>	36
6.4.2	We look at <i>soil.fac</i>	37
6.4.3	We look at <i>well.fac</i>	38

6.4.4	We look at <i>depth</i>	39
6.4.5	We look at <i>seed.weight.grams</i>	40
6.4.6	We look at <i>broken</i>	41
6.4.7	We look at <i>germinated.in.lab</i>	42
6.4.8	We look at <i>germinated.in.field</i>	43
6.4.9	We look at the coordinates	44
6.5	We look at the position in the field	46
6.6	Response variable: <i>days.to.germination</i>	48
6.6.1	We look at <i>pot.fac</i>	48
6.6.2	We look at <i>soil.fac</i>	49
6.6.3	We look at <i>well.fac</i>	50
6.6.4	We look at <i>seed.weight.grams</i>	51
6.6.5	We look at <i>depth</i>	52
6.6.6	We look at <i>broken</i>	53
6.6.7	We look at the coordinates	54
7	Session information	57

1 Freezing Package versions

```
## (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(lubridate)
library(stringr)
library(ggbeeswarm)
library(tableone)
library(tidyr)
library(mgcv)
library(survival)
library(survminer)
```

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 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 Table 1

The following list of variables are neither numeric nor character variables and are not shown in the Table 1.

```

d.maize %>%
  summarize(across(.cols = everything(), .fns = class)) %>%
  pivot_longer(cols = everything()) %>%
  filter( ! (value %in% c("numeric", "integer", "factor", "character"))) %>%
  pull(name)

```

```

[1] "date.germinated.asDate" "broken"          "plant.found"
[4] "germinated.in.lab"     "germinated.in.field" "germinated.yes"

```

5.1 Categorical variables

We only present categorical variables with less than or equal to 10 different categories.

```

## get categorical variables
vars.cat <- lapply(X = d.maize, FUN = function(x){
  class(x) %in% c("character", "factor")
}) %>%
  unlist()
## get variables with <= 10 different categories
vars.cat.10 <- d.maize %>%
  select(all_of(names(vars.cat)[vars.cat])) %>%
  apply(MARGIN = 2, FUN = function(x){
    n_distinct(x, na.rm = TRUE) <= 10
  })
vars.cat.10 <- names(vars.cat.10)[vars.cat.10]

```

The following list of categorical variables are not shown in the Table 1 (more than 10 levels).

```

## categorical variables not presented in Table 1 (more than 10 levels):
vars.cat.not10 <- setdiff(names(vars.cat)[vars.cat], vars.cat.10)
vars.cat.not10

```

```

[1] "pot"          "height_2022_07_05" "cob_weight"
[4] "pot.fac"

```

```

vars.cont <- d.maize %>%
  select(where(is.numeric)) %>%
  colnames()
vars.tab1 <- vars.cat.10
vars.tab1 <- colnames(d.maize)[colnames(d.maize) %in% vars.tab1]

```

We now calculate the Table 1.

```
## (output and warnings are omitted in this chunk)
##
## calculate table 1
tab1 <- CreateTableOne(
  vars = as.character(vars.tab1),
  data = d.maize[, colnames(d.maize) %in% vars.tab1],
  test = FALSE)
tab1.print <- tableone:::print.TableOne(x = tab1,
  showAllLevels = FALSE,
  nonnormal = TRUE,
  dropEqual = FALSE,
  minMax = FALSE,
  explain = TRUE,
  missing = TRUE)
tab1.print <- tableone:::print.TableOne(x = tab1,
  showAllLevels = FALSE,
  nonnormal = TRUE,
  dropEqual = FALSE,
  minMax = FALSE,
  explain = TRUE,
  missing = TRUE)
tab1.res_indent <- seq_along(rownames(tab1.print))
tab1.res_indent <- tab1.res_indent[str_detect(string = rownames(tab1.print),
  pattern = "  ")]
```

We now print the Table 1.

```
## print table
tab1.print %>%
  as_tibble(rownames = " ") %>%
  mutate(` ` = str_sub(string = ` ` , start = 1, end = 70)) %>%
  kable(caption = paste0("Descriptive statistics for categorical variables: ",
    "frequencies and percentages in brackets are displayed. ",
    "Only one category is shown for binary variables."),
    label = "tab1_categorical",
    booktabs=TRUE,
    longtable = TRUE,
    linesep = c(" ")) %>%
  # landscape() %>%
  #column_spec(column = 1, width = "5in") %>%
  kable_styling(font_size = 7,
    latex_options = c("striped", "repeat_header", "hold_position")) %>%
  add_indent(tab1.res_indent)
```

Table 1: Descriptive statistics for categorical variables: frequencies and percentages in brackets are displayed. Only one category is shown for binary variables.

	Overall	Missing
n	108	
soil (%)		0.0
Bio garden	24 (22.2)	
Composana	24 (22.2)	
herbs	36 (33.3)	
mixture	24 (22.2)	
well (%)		0.0
a	18 (16.7)	
b	18 (16.7)	

Table 1: Descriptive statistics for categorical variables: frequencies and percentages in brackets are displayed. Only one category is shown for binary variables. *(continued)*

	Overall	Missing
c	18 (16.7)	
d	18 (16.7)	
e	18 (16.7)	
f	18 (16.7)	
fungus = yes (%)	1 (100.0)	99.1
date.germinated (%)		22.2
2022-05-07	4 (4.8)	
2022-05-08	1 (1.2)	
2022-05-09	6 (7.1)	
2022-05-10	17 (20.2)	
2022-05-11	29 (34.5)	
2022-05-12	13 (15.5)	
2022-05-13	10 (11.9)	
2022-05-14	4 (4.8)	
observations (%)		93.5
Broken	3 (42.9)	
Broken, morning measurement	1 (14.3)	
morning measurement	3 (42.9)	
soil.fac (%)		0.0
Bio garden	24 (22.2)	
Composana	24 (22.2)	
herbs	36 (33.3)	
mixture	24 (22.2)	
well.fac (%)		0.0
a	18 (16.7)	
b	18 (16.7)	
c	18 (16.7)	
d	18 (16.7)	
e	18 (16.7)	
f	18 (16.7)	
fungus.fac = yes (%)	1 (0.9)	0.0
obs.time = night (%)	104 (96.3)	0.0

```
write.csv2(x = tab1.print,
  file = paste0("./Prepared_data_and_models/", "d.maize_table1.csv"),
  fileEncoding = "UTF-8",
  quote = TRUE,
  row.names = TRUE)
```

5.2 Continuous variables

```
## calculate summary statistics
tab1.print.cont <- d.maize %>%
  select(where(is.numeric)) %>%
  tidyr::pivot_longer(cols = select(.data = d.maize,
    where(is.numeric),
    -any_of("")), %>%
    colnames()) %>%
  mutate(name = factor(x = name, levels = unique(name))) %>%

## group_by (variable and strata)
group_by_at(c("name",
  select(.data = d.maize,
    any_of(c("name", ""))) %>%
    colnames())) %>%
  summarize(
    mean = mean(value, na.rm = TRUE),
```

```

sd = sd(value, na.rm = TRUE),
median = median(value, na.rm = TRUE),
quartiles.25 = quantile(x = value, probs = 0.25, na.rm = TRUE),
quartiles.75 = quantile(x = value, probs = 0.75, na.rm = TRUE),
IQR = IQR(value, na.rm = TRUE),
n = sum(is.na(value)),
n.perc = n / n() * 100, .groups = "keep") %>%
ungroup() %>%

## rounding
mutate(n = format(x = round(x = n, digits = 1),
                    nsmall = 0)) %>%
mutate(n.perc = format(x = round(x = n.perc, digits = 1),
                             nsmall = 1)) %>%
mutate(across(.cols = c(mean, sd, median, quartiles.25, quartiles.75, IQR), .fns = function(x){
  signif(x = x, digits = 3) %>%
  as.character()
})) %>%
mutate("Mean (SD)" = paste0(mean, " (", sd, ")")) %>%
mutate("Median [IQR]" = paste0(median, " [", quartiles.25, ", ", quartiles.75, "]")) %>%
mutate("Missings" = paste0(n, " (", n.perc, "%)") %>%
select(name, any_of(c("Mean (SD)", "Median [IQR]", "Missings"))))

## print table
tbl.print.cont %>%
mutate(`name` = str_sub(string = `name`, start = 1, end = 70)) %>%
kable(caption = paste0("Descriptive statistics for continuous variables: ",
                        "mean and standard deviation (SD) as well as median ",
                        "and interquartile range (IQR) are presented."),
      label = "tbl_continuous",
      booktabs=TRUE,
      longtable = TRUE,
      linesep = c("")) %>%
# landscape() %>%
# column_spec(column = 1, width = "2.8in") %>%
kable_styling(font_size = 7,
              latex_options = c("striped", "repeat_header", "hold_position"))

```

Table 2: Descriptive statistics for continuous variables: mean and standard deviation (SD) as well as median and interquartile range (IQR) are presented.

name	Mean (SD)	Median [IQR]	Missings
depth	3.5 (1.72)	3.5 [2, 5]	0 (0.0%)
seed.weight	29.4 (5.95)	29 [25, 34]	0 (0.0%)
...12	149 (NA)	149 [149, 149]	107 (99.1%)
seed.weight.grams	0.294 (0.0595)	0.29 [0.25, 0.34]	0 (0.0%)
height_2022_07_05.num	220 (48.8)	235 [204, 250]	48 (44.4%)
cob_weight.num	117 (49)	117 [98.5, 142]	28 (25.9%)
days.to.germination	11 (1.56)	11 [10, 12]	24 (22.2%)
days.to.germination.censored	11.6 (1.87)	11 [10, 14]	0 (0.0%)
seed_coord_y	5 (2.59)	5 [3, 7]	0 (0.0%)
seed_coord_x	6.5 (3.47)	6.5 [3.75, 9.25]	0 (0.0%)
position_field_x	2.86 (1.41)	3 [2, 4]	0 (0.0%)
position_field_x_cm	143 (70.5)	150 [100, 200]	0 (0.0%)
position_field_y	11.4 (6.45)	11 [6, 17]	0 (0.0%)
position_field_y_cm	285 (161)	275 [150, 425]	0 (0.0%)

```
write.csv2(x = tab1.print.cont,
  file = paste0("./Prepared_data_and_models/",
    "d.maize_table1.continuous.csv"),
  fileEncoding = "UTF-8",
  quote = TRUE,
  row.names = FALSE)
```

6 Graphical Analysis

We first create ‘proto-plot’ functions. One will be used for continuous predictors while the other two for categorical predictors.

```
f.gg.scatter <- function(data, x, y, group = NULL,
  fill = NULL, color = NULL,
  plot.alpha = 1,
  ...){

  if(!missing(x)){x <- enquos(x)}else{x = NULL}
  if(!missing(y)){y <- enquos(y)}else{y = NULL}
  if(!missing(group)){group <- enquos(group)}else{group = NULL}
  if(!missing(fill)){fill <- enquos(fill)}else{fill = NULL}
  if(!missing(color)){color <- enquos(color)}else{color = NULL}

  ## create ggplot object
  gg.ret <- data %>%
    ggplot() +
    aes(x = !!x, y = !!y, color = !!color, fill = !!fill, group = !!group) +

    geom_point(alpha = plot.alpha) +
    geom_smooth(method = "loess", formula = "y ~ x", se = TRUE)

  ## project-specific tailoring
  gg.ret <- gg.ret +
    labs()

  return(gg.ret)
}
```

```
f.gg.density <- function(data, x, y, group = NULL,
  fill = NULL, color = NULL,
  plot.alpha = 1,
  ...){

  if(!missing(x)){x <- enquos(x)}else{x = NULL}
  if(!missing(y)){y <- enquos(y)}else{y = NULL}
  if(!missing(group)){group <- enquos(group)}else{group = NULL}
  if(!missing(fill)){fill <- enquos(fill)}else{fill = NULL}
  if(!missing(color)){color <- enquos(color)}else{color = NULL}

  ## create ggplot object
  gg.ret <- data %>%
    ggplot() +
```

```

    ## to not drop empty factor levels
    scale_color_discrete(drop = FALSE) +
    scale_fill_discrete(drop = FALSE) +

    aes(y = !!y, color = !!color, fill = !!fill, group = !!group) +
    geom_density(alpha = plot.alpha) +
    geom_rug(alpha = plot.alpha) +
    coord_flip()

    ## project-specific tailoring
    gg.ret <- gg.ret +
      labs()

    return(gg.ret)
}

f.gg.boxplot <- function(data, x, y,
                        fill = NULL, color = NULL,
                        group = NULL,
                        plot.alpha = 1,
                        plot.beeswarm = FALSE,
                        ...){

  if(!missing(x)){x <- enquo(x)}else{x = NULL}
  if(!missing(y)){y <- enquo(y)}else{y = NULL}
  if(!missing(group)){group <- enquo(group)}else{group = NULL}
  if(!missing(fill)){fill <- enquo(fill)}else{fill = NULL}
  if(!missing(color)){color <- enquo(color)}else{color = NULL}

  ## create ggplot object
  gg.ret <- data %>%
    ggplot() +

    ## to not drop empty factor levels
    scale_color_discrete(drop = FALSE) +
    scale_fill_discrete(drop = FALSE) +

    aes(x = !!x, y = !!y, color = !!color, fill = !!fill, group = !!group)

  ## use beeswarm or not
  if(plot.beeswarm){
    gg.ret <- gg.ret +
      geom_boxplot(outlier.alpha = 0) +
      geom_beeswarm(groupOnX = FALSE, alpha = plot.alpha)
  }else{
    gg.ret <- gg.ret +
      geom_boxplot(outlier.alpha = plot.alpha, alpha = plot.alpha)
  }

  ## project-specific tailoring
  gg.ret <- gg.ret +
    labs()

```

```

    return(gg.ret)
}

```

The subsequent overall function is used to call the corresponding plot function of the correct type.

```

f.gg.proto.builder <- function(data, x, y = NULL,
                                type,
                                na.rm.pairwise = TRUE,
                                ...){

  ## needed for filter "na.rm.pairwise"
  if(!missing(x)){x <- enquo(x)}else{x = NULL}
  if(!missing(y)){y <- enquo(y)}else{y = NULL}

  ## remove missing values pairwise
  if(na.rm.pairwise & !is.null(x)){
    removed.rows <- data %>%
      filter(is.na(!x) | is.na(!y)) %>%
      nrow()
    if(removed.rows > 0){
      message(paste0("(m): Deleted ", removed.rows,
                      " row(s) due to pairwise missing data. "))
      data <- data %>%
        filter(!is.na(!x) & !is.na(!y))
    }
  }

  ## plot type
  if(type == "scatter"){
    gg.out <- f.gg.scatter(data = data, x = !!x, y = !!y, ...)
  }
  if(type == "density"){
    gg.out <- f.gg.density(data = data, y = !!y, ...)
  }
  if(type == "boxplot"){
    gg.out <- f.gg.boxplot(data = data, x = !!x, y = !!y, ...)
  }

  return(gg.out)
}

```

The next function is used to set default values for all plots.

```

## setting defaults
f.gg.proto <- function(...){
  f.gg.proto.builder(data = d.maize,
                      plot.beeswarm = TRUE,
                      na.rm.pairwise = TRUE,
                      plot.alpha = 0.3,
                      ... )
}

```

6.1 Response variable: *cob_weight.num*

6.1.1 We look at *pot.fac*

```
f.gg.proto(type = "boxplot", y = `cob_weight.num`, x = `pot.fac`) +  
  scale_x_discrete(drop = FALSE) +  
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

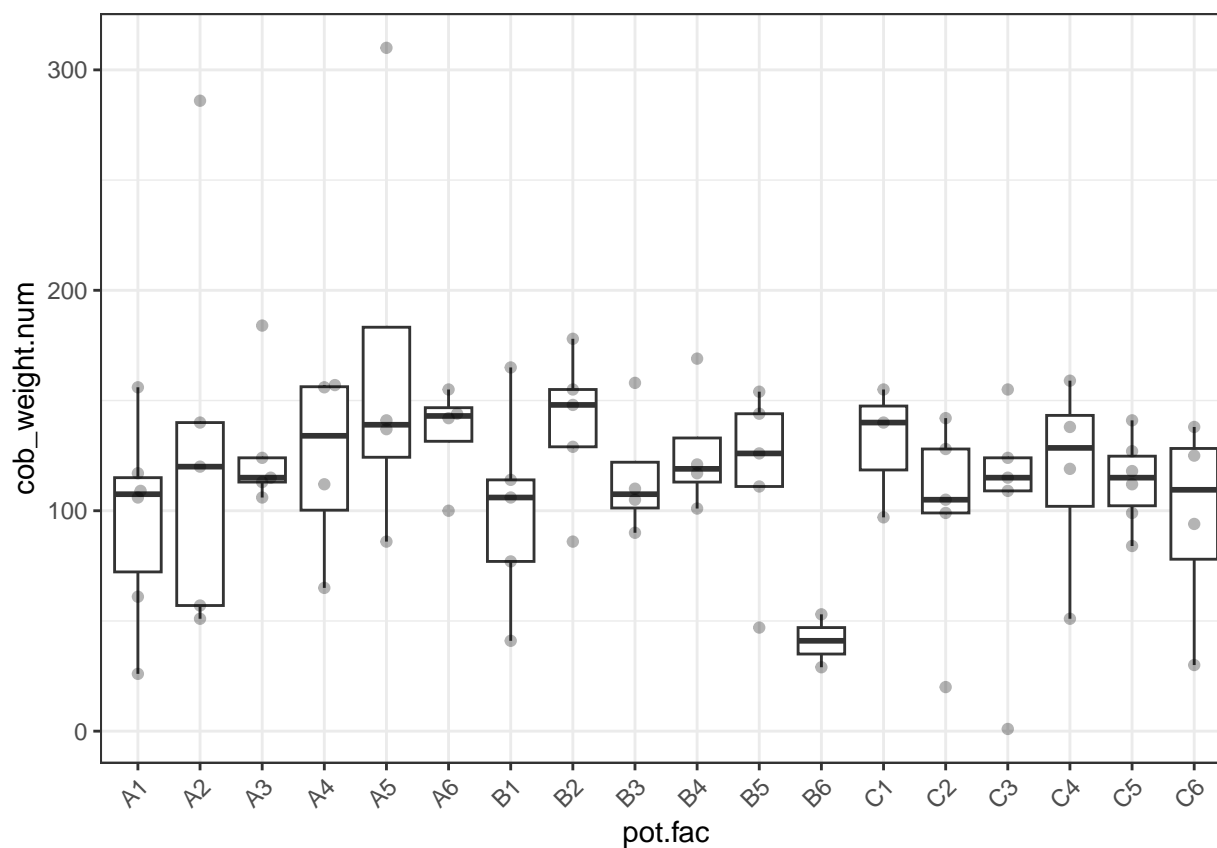
(m): Deleted 28 row(s) due to pairwise missing data.

Warning: The `groupOnX` argument of `geom_beeswarm()` is deprecated as of ggbeeswarm 0.7.1.

i ggplot2 now handles this case automatically.

This warning is displayed once every 8 hours.

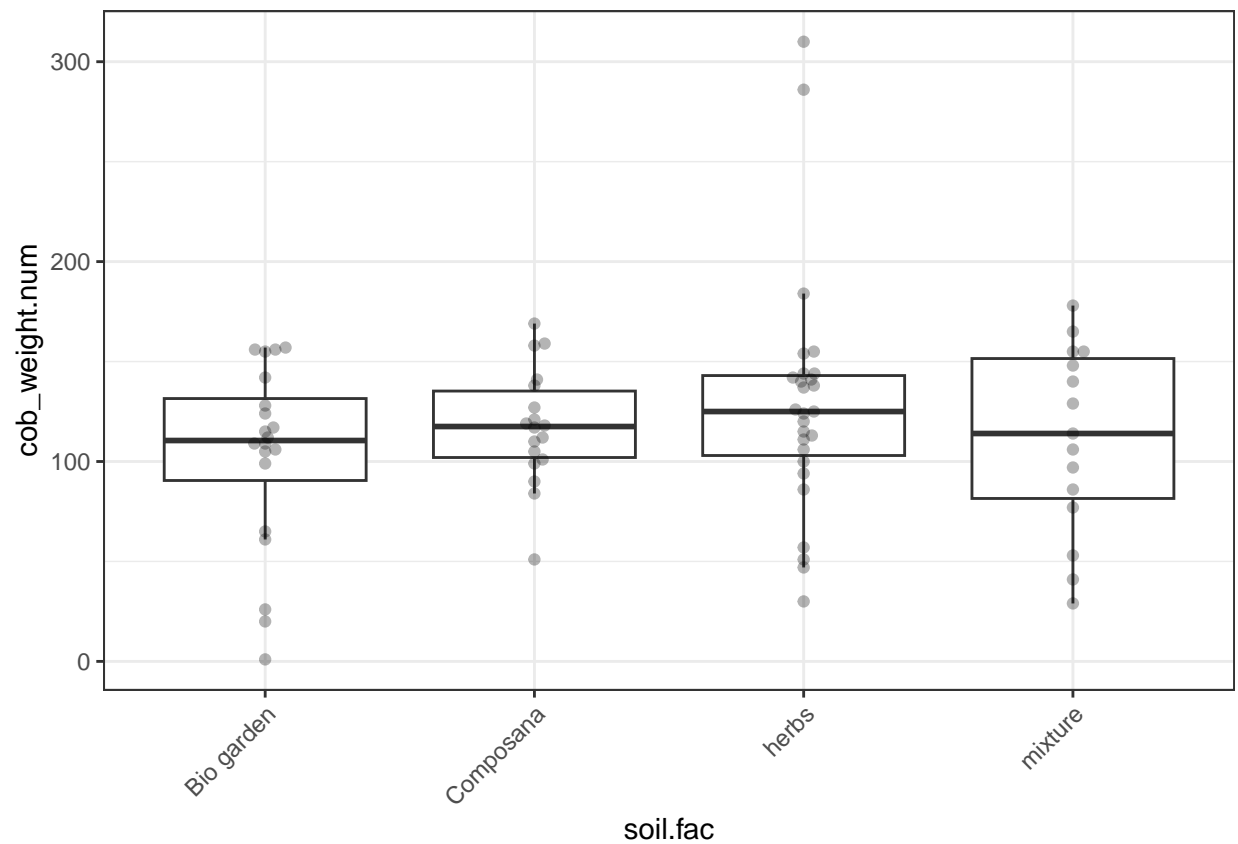
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.



6.1.2 We look at *soil.fac*

```
f.gg.proto(type = "boxplot", y = `cob_weight.num`, x = `soil.fac`) +  
  scale_x_discrete(drop = FALSE) +  
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

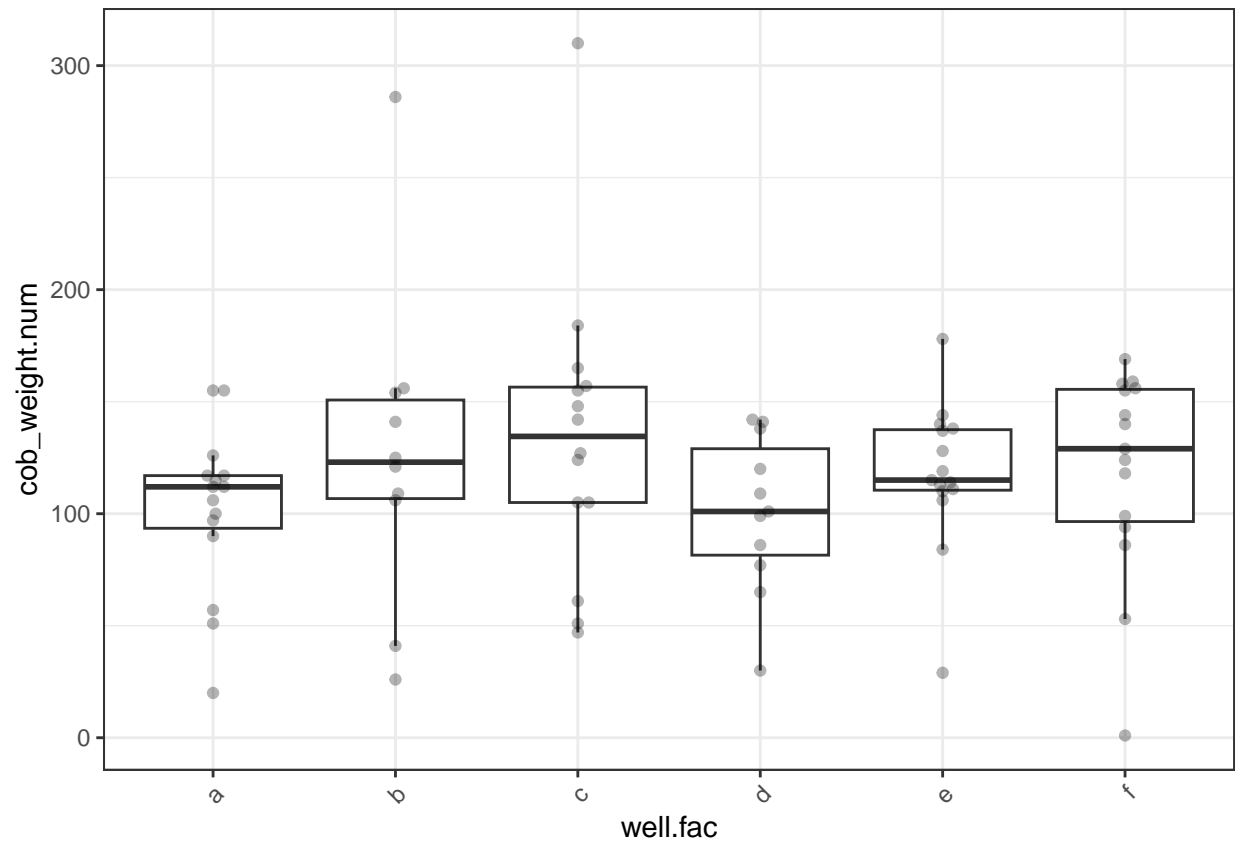
(m): Deleted 28 row(s) due to pairwise missing data.



6.1.3 We look at *well.fac*

```
f.gg.proto(type = "boxplot", y = `cob_weight.num`, x = `well.fac`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

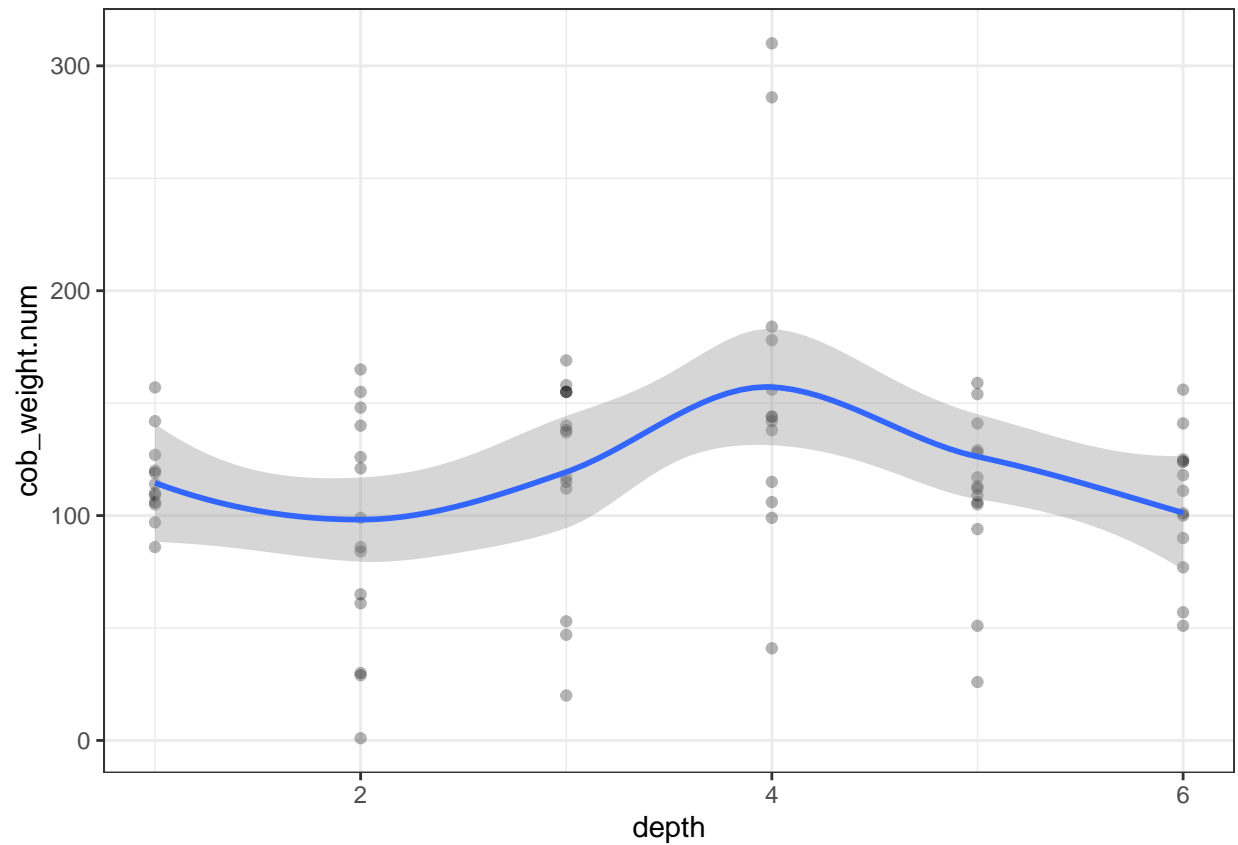
(m): Deleted 28 row(s) due to pairwise missing data.



6.1.4 We look at *depth*

```
f.gg.proto(type = "scatter", y = `cob_weight.num`, x = `depth`)
```

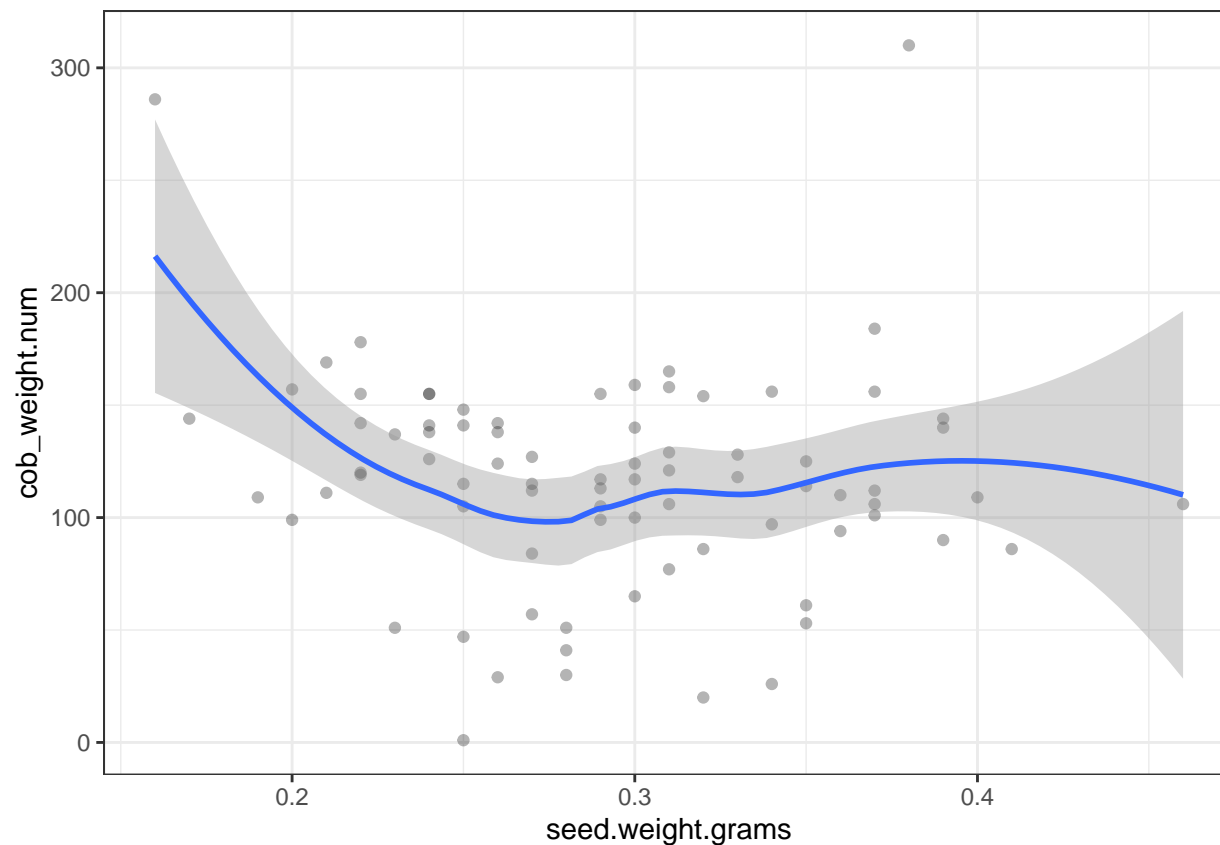
(m): Deleted 28 row(s) due to pairwise missing data.



6.1.5 We look at *seed.weight.grams*

```
f.gg.proto(type = "scatter",
  y = `cob_weight.num`,
  x = `seed.weight.grams`)
```

(m): Deleted 28 row(s) due to pairwise missing data.



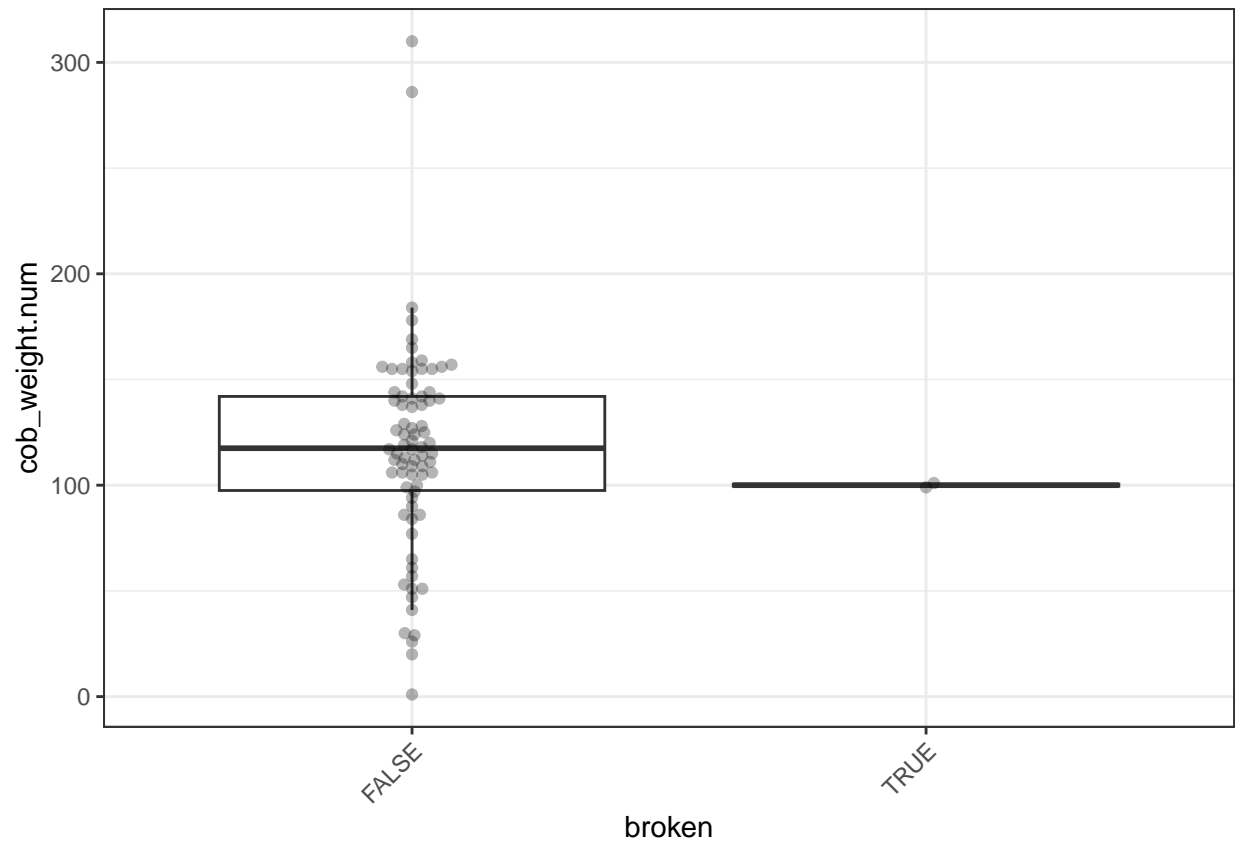
6.1.6 We look at *broken*

```
table(d.maize$broken, useNA = "ifany")
```

```
FALSE  TRUE
  104     4
```

```
f.gg.proto(type = "boxplot", y = `cob_weight.num`, x = `broken`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

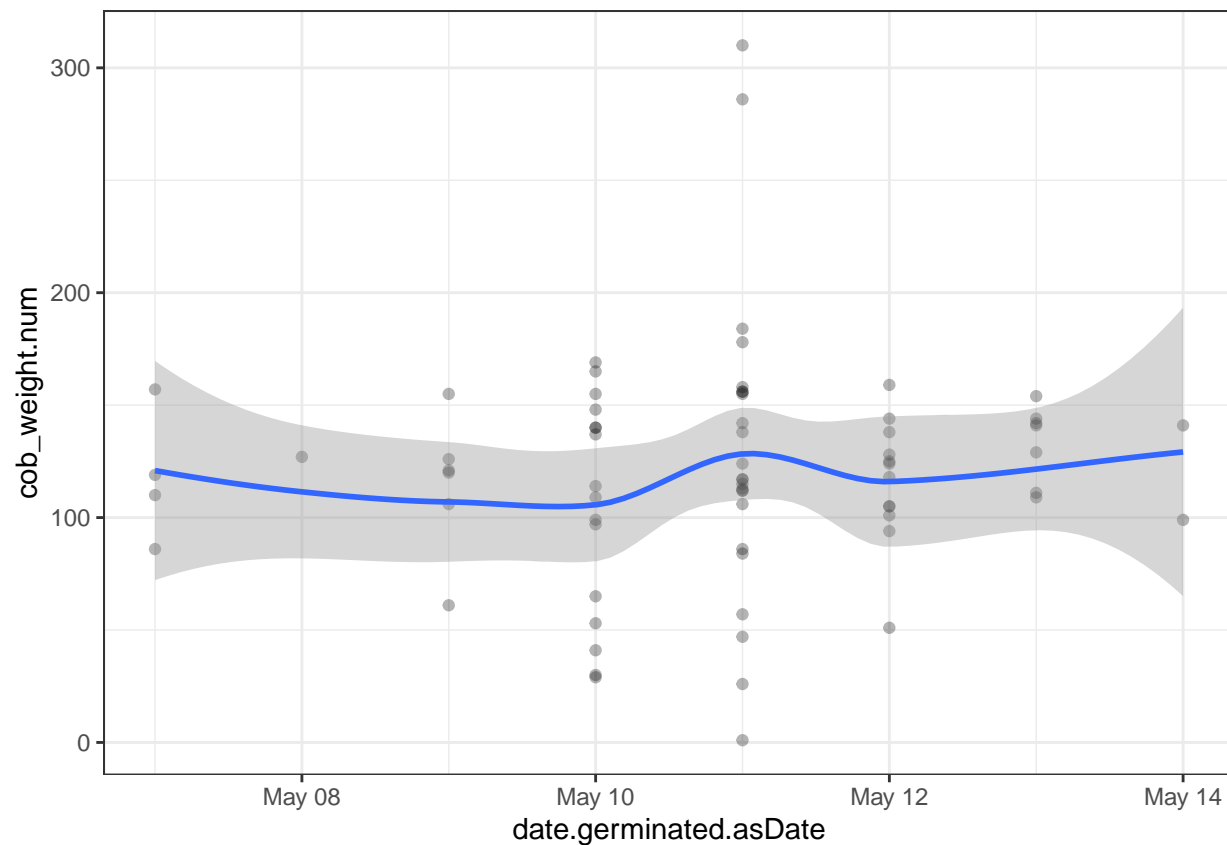
(m): Deleted 28 row(s) due to pairwise missing data.



6.1.7 We look at *date.germinated.asDate*

```
f.gg.proto(type = "scatter",  
  y = `cob_weight.num`,  
  x = `date.germinated.asDate`)
```

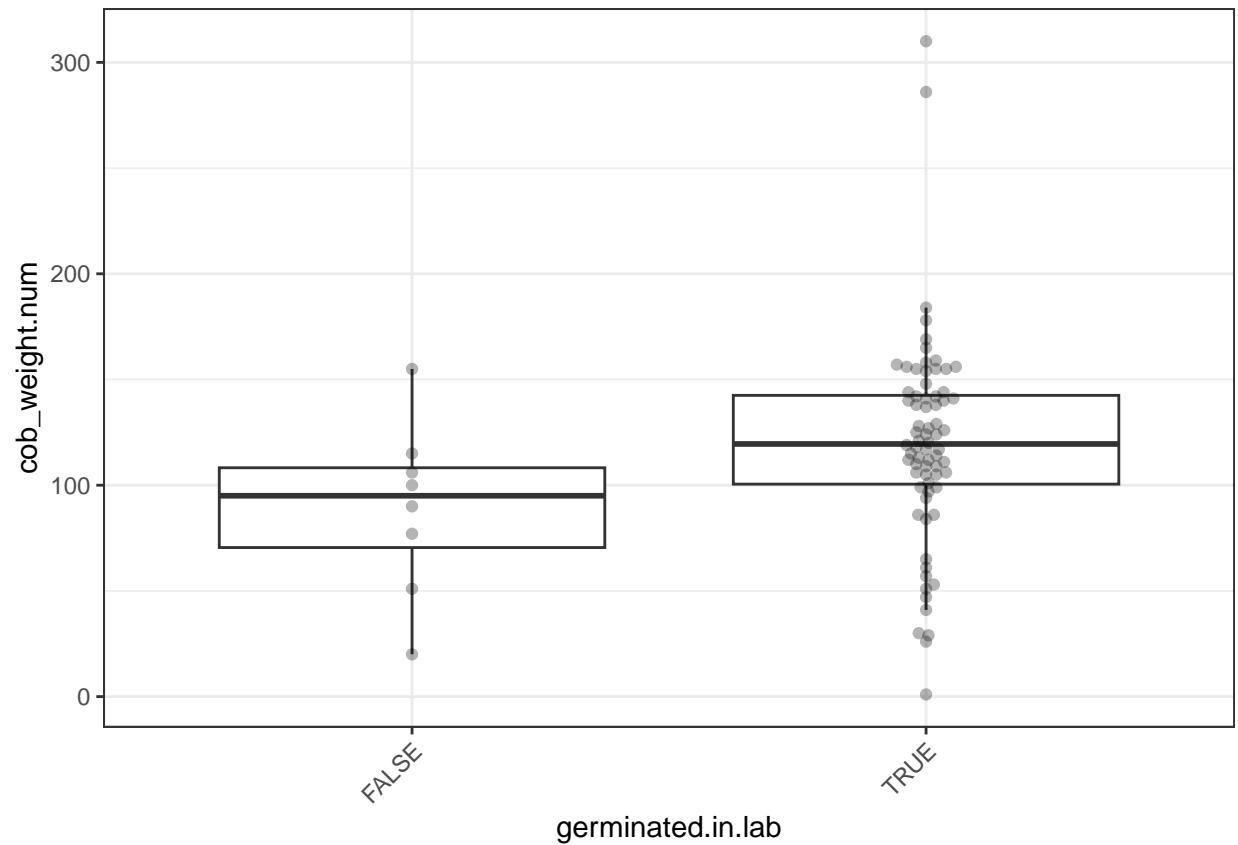
(m): Deleted 36 row(s) due to pairwise missing data.



6.1.8 We look at *germinated.in.lab*

```
f.gg.proto(type = "boxplot", y = `cob_weight.num`, x = `germinated.in.lab`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

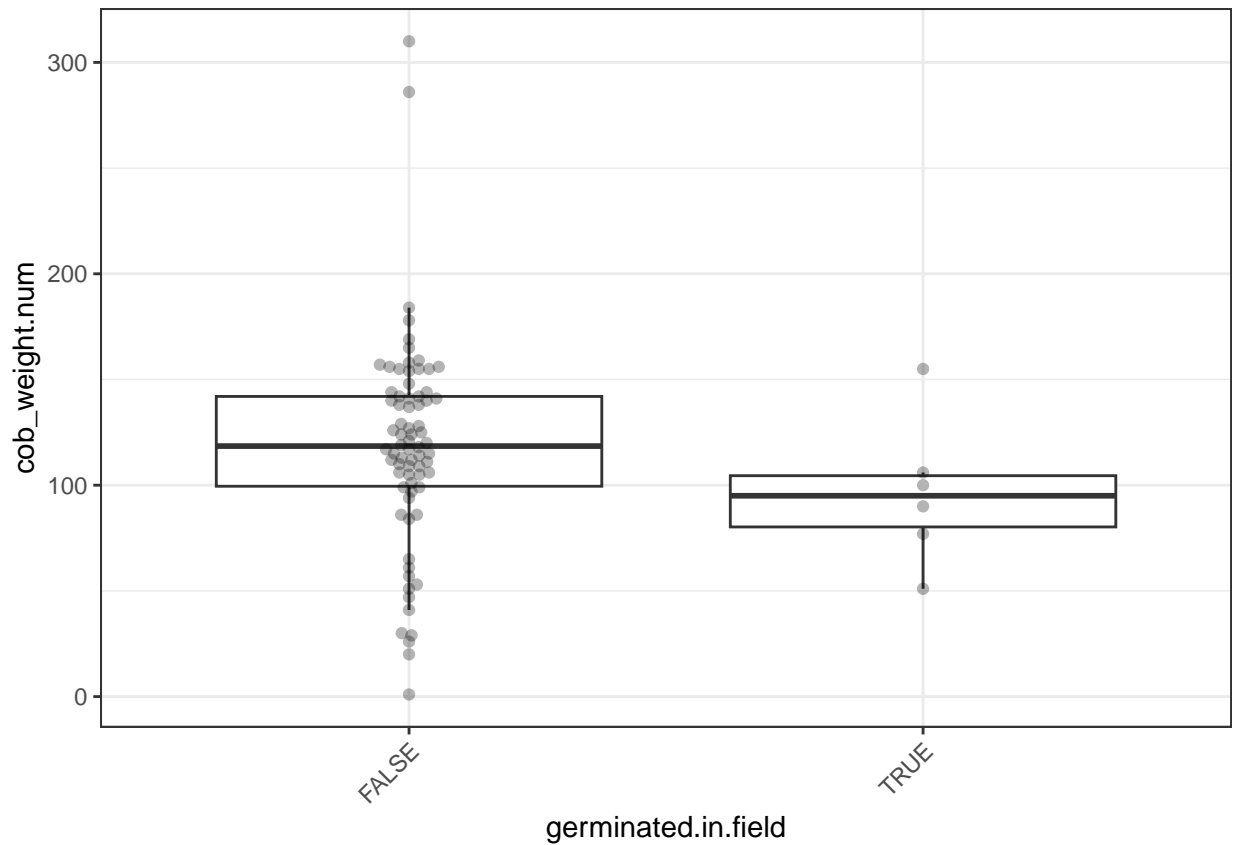
(m): Deleted 28 row(s) due to pairwise missing data.



6.1.9 We look at *germinated.in.field*

```
f.gg.proto(type = "boxplot", y = `cob_weight.num`, x = `germinated.in.field`) +  
  scale_x_discrete(drop = FALSE) +  
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

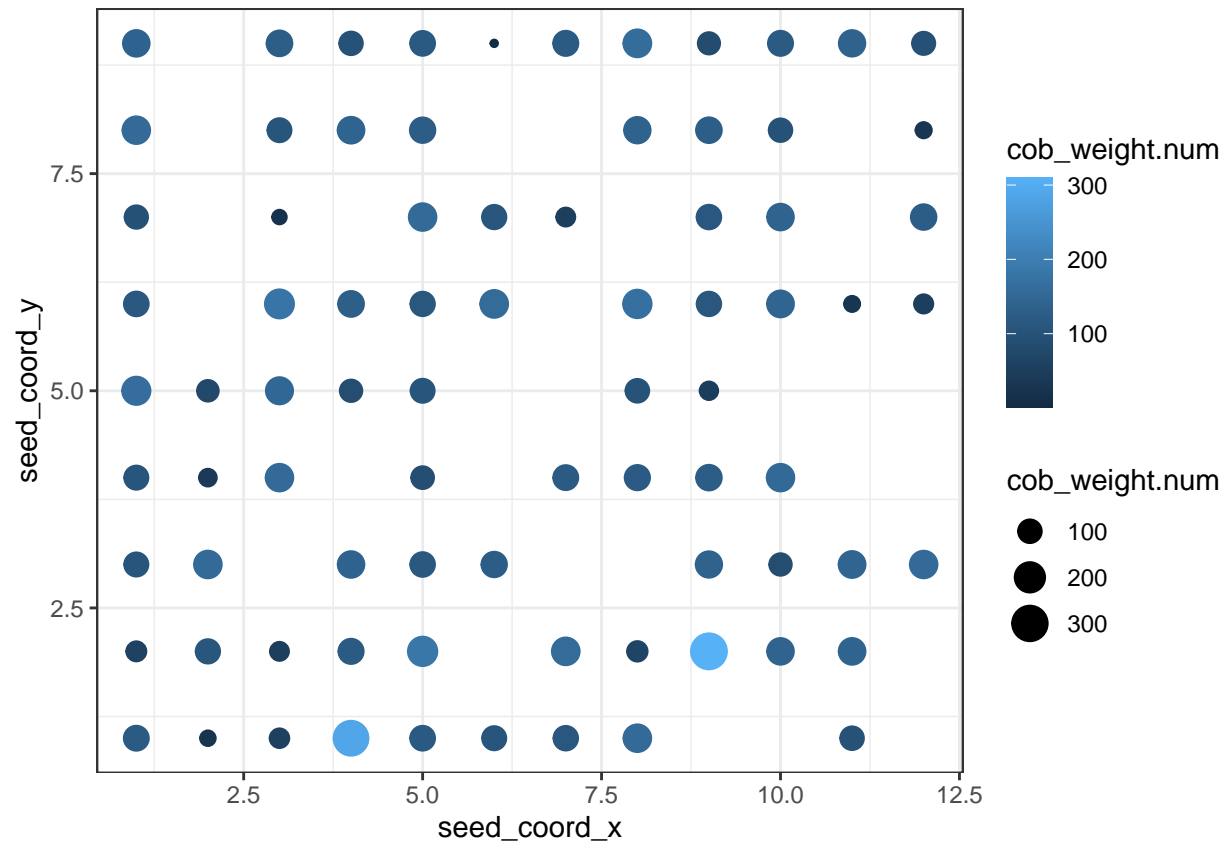
(m): Deleted 28 row(s) due to pairwise missing data.



6.1.10 We look at the coordinates

```
ggplot(d.maize, mapping = aes(x = seed_coord_x, y = seed_coord_y)) +  
  geom_point(mapping = aes(size = cob_weight.num,  
                           colour = cob_weight.num))
```

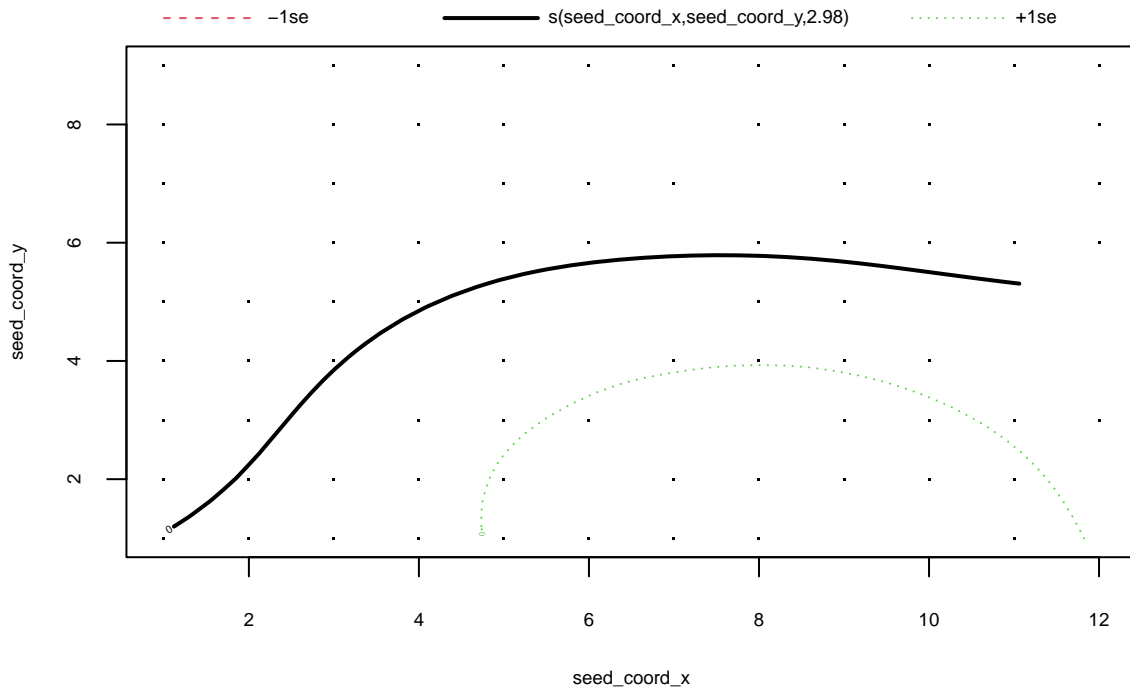
Warning: Removed 28 rows containing missing values (`geom_point()`).



The bigger and light the point on the plot, the larger the weight of the cob is.

There doesn't appear to be a clear pattern of cob weight depending on the lab position.

```
gam.cob_weight.coord <- gam(cob_weight.num ~ s(seed_coord_x,
                                              seed_coord_y,
                                              k = 30),
                           data = d.maize)
plot(gam.cob_weight.coord)
```

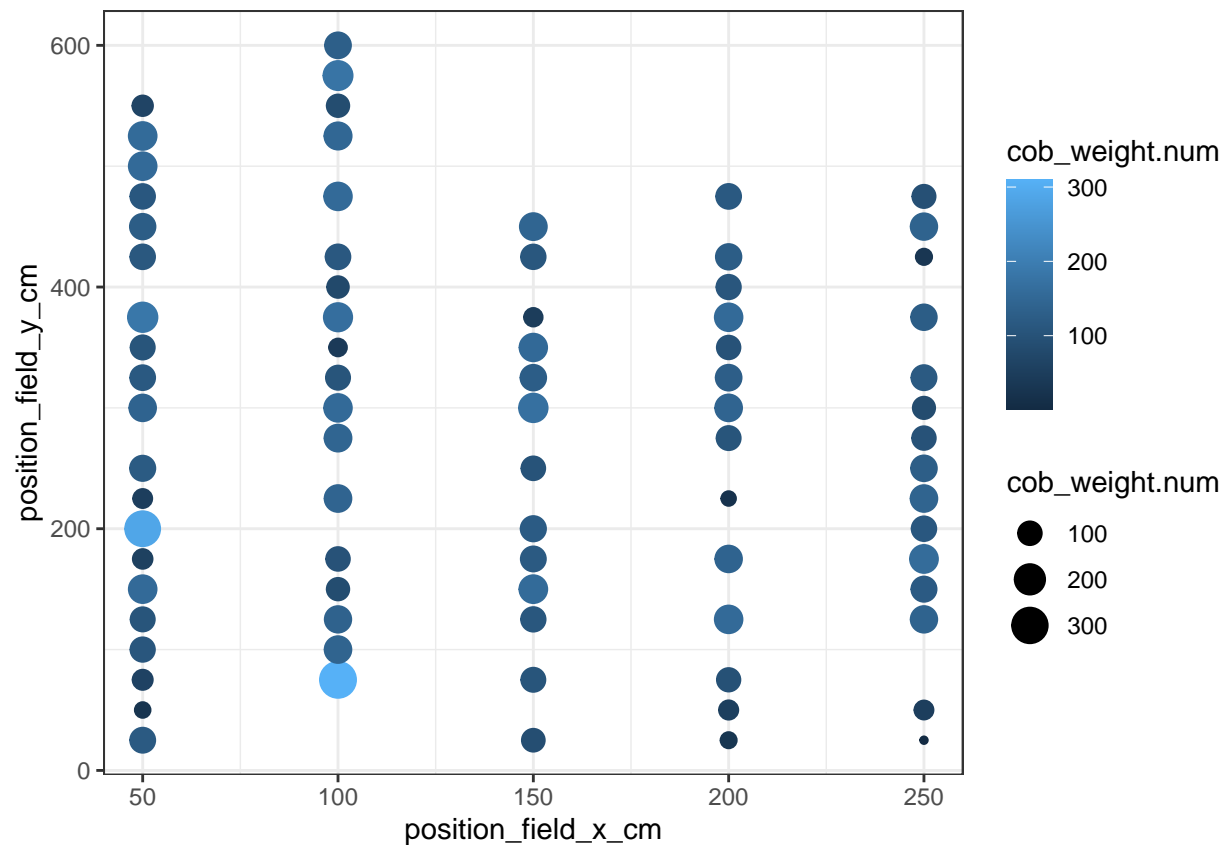


This plot appears to confirm the previously mentioned observation that there isn't a strong correlation between cob weight and lab position. Indeed, we can observe a black line representing a zero gradient, and the standard errors of 1 and -1 deviate significantly from this line. This demonstrates that there is no clear indication of a consistent pattern of cob weight depending on the lab position.

6.1.11 We look at the position in the field

```
ggplot(d.maize, mapping = aes(x = position_field_x_cm, y = position_field_y_cm)) +
  geom_point(mapping = aes(size = cob_weight.num,
                           colour = cob_weight.num))
```

Warning: Removed 28 rows containing missing values (`geom_point()`).

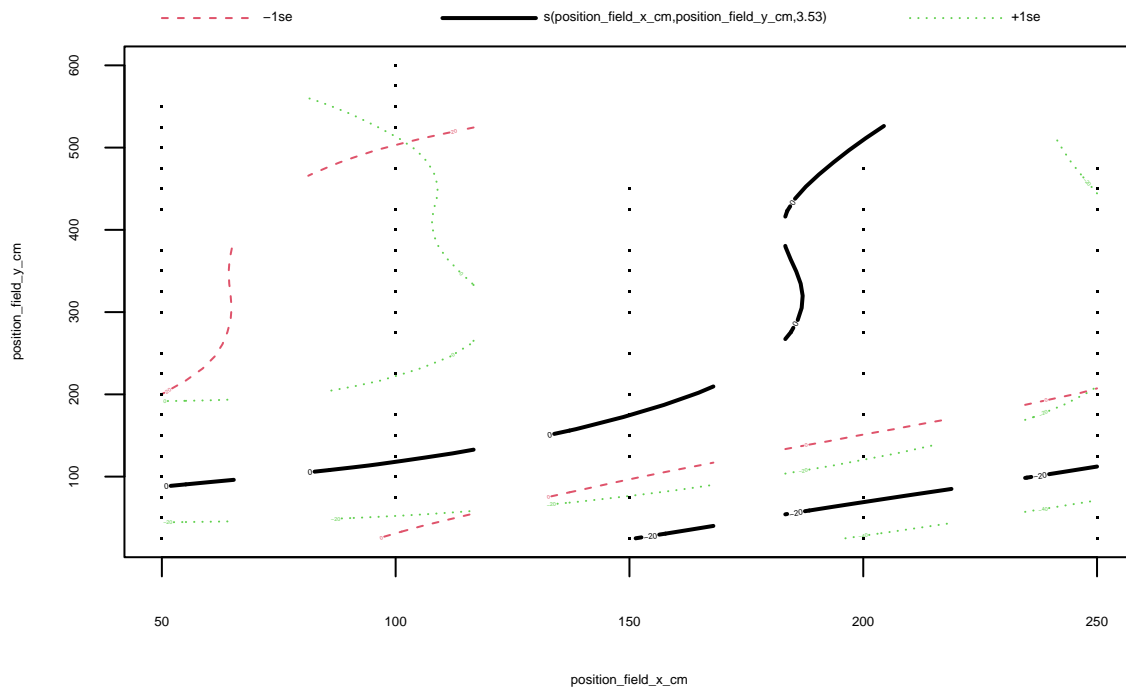


The bigger and light the point on the plot, the larger the weight of the cob is.

There seems to be a slight difference between cob weight depending on the field position.

Indeed, as shown also in the graph below, the plants on the left-upper maizeer seem to have a slightly larger cob weight than the plants positioned on the right-bottom maizeer.

```
gam.cob_weight.field <- gam(cob_weight.num ~ s(position_field_x_cm,
                                              position_field_y_cm,
                                              k = 30),
                           data = d.maize)
plot(gam.cob_weight.field)
```

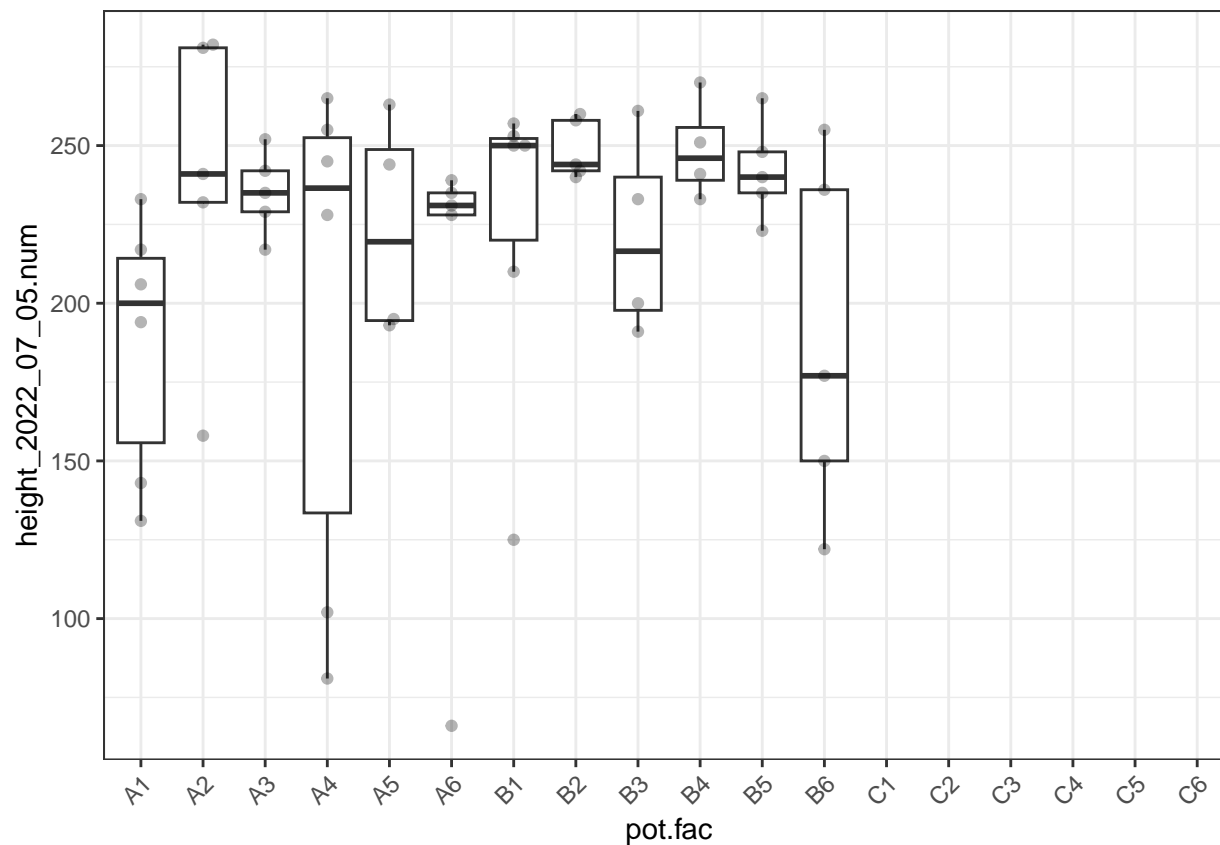


6.2 Response variable: *height_2022_07_05.num*

6.2.1 We look at *pot.fac*

```
f.gg.proto(type = "boxplot", y = `height_2022_07_05.num`, x = `pot.fac`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

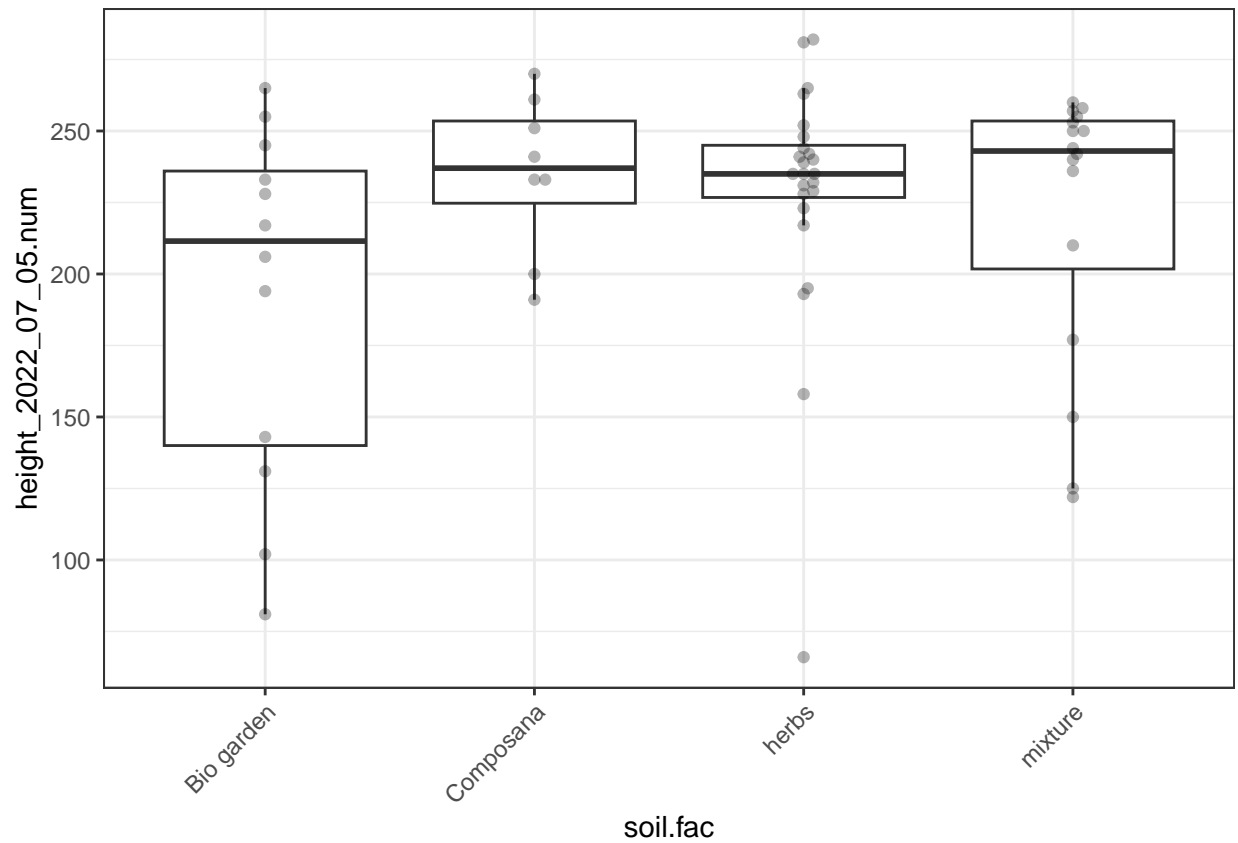
(m): Deleted 48 row(s) due to pairwise missing data.



6.2.2 We look at *soil.fac*

```
f.gg.proto(type = "boxplot", y = `height_2022_07_05.num`, x = `soil.fac`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

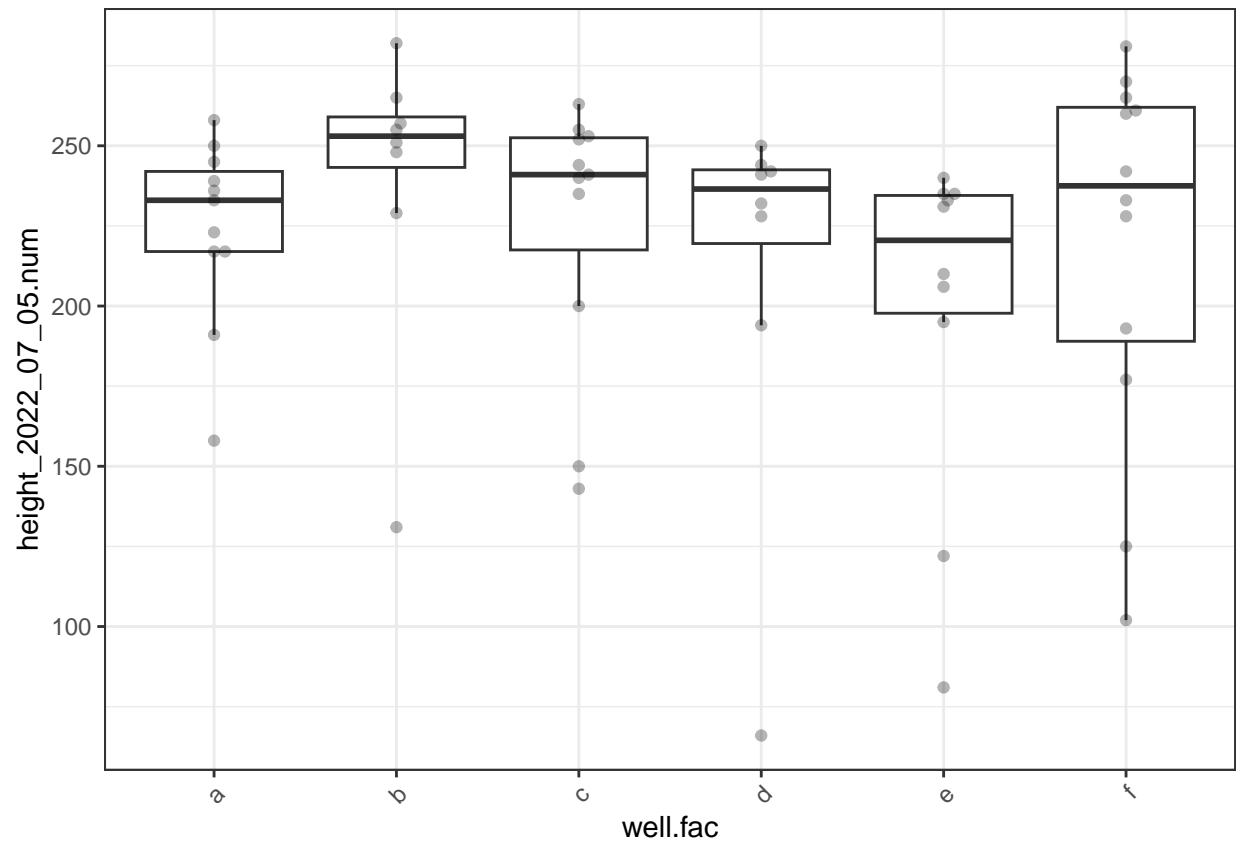
(m): Deleted 48 row(s) due to pairwise missing data.



6.2.3 We look at *well.fac*

```
f.gg.proto(type = "boxplot", y = `height_2022_07_05.num`, x = `well.fac`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

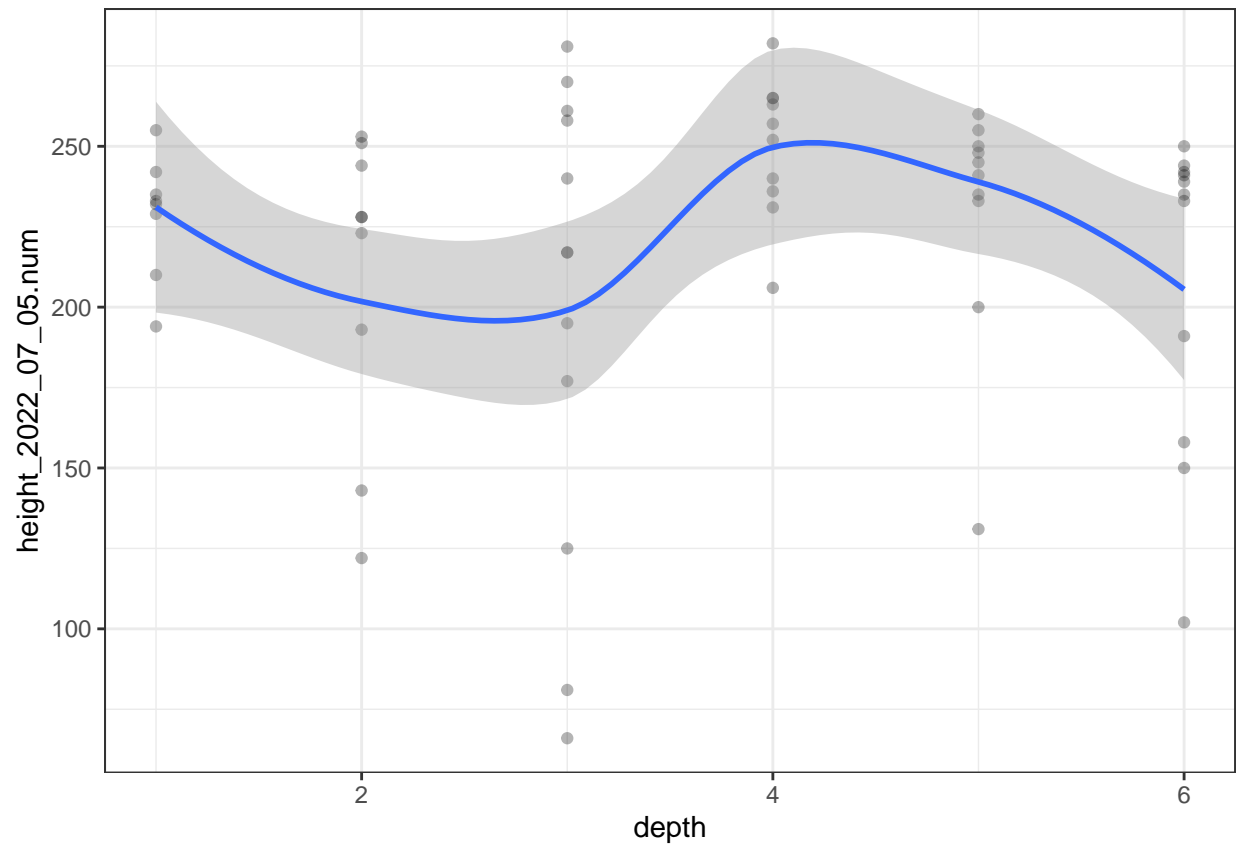
(m): Deleted 48 row(s) due to pairwise missing data.



6.2.4 We look at *depth*

```
f.gg.proto(type = "scatter", y = `height_2022_07_05.num`, x = `depth`)
```

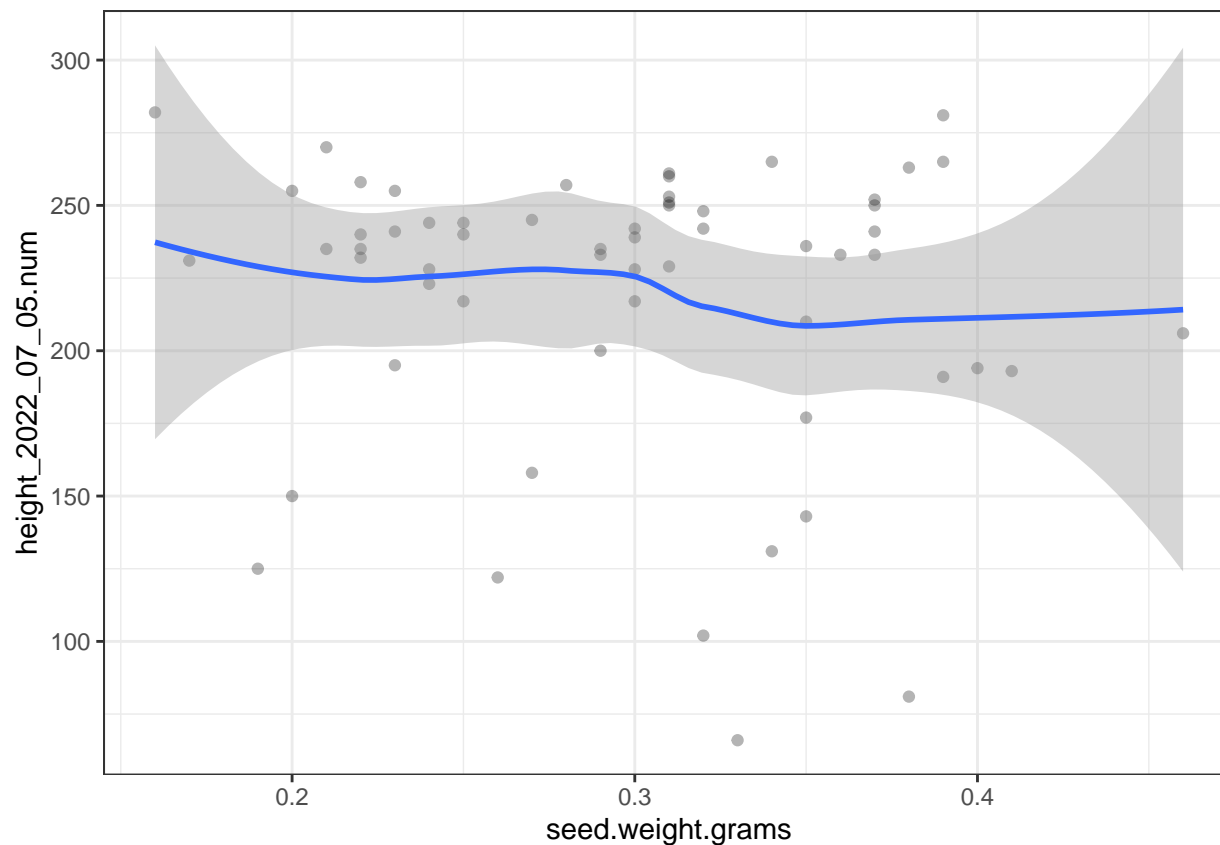
(m): Deleted 48 row(s) due to pairwise missing data.



6.2.5 We look at *seed.weight.grams*

```
f.gg.proto(type = "scatter", y = `height_2022_07_05.num`, x = `seed.weight.grams`)
```

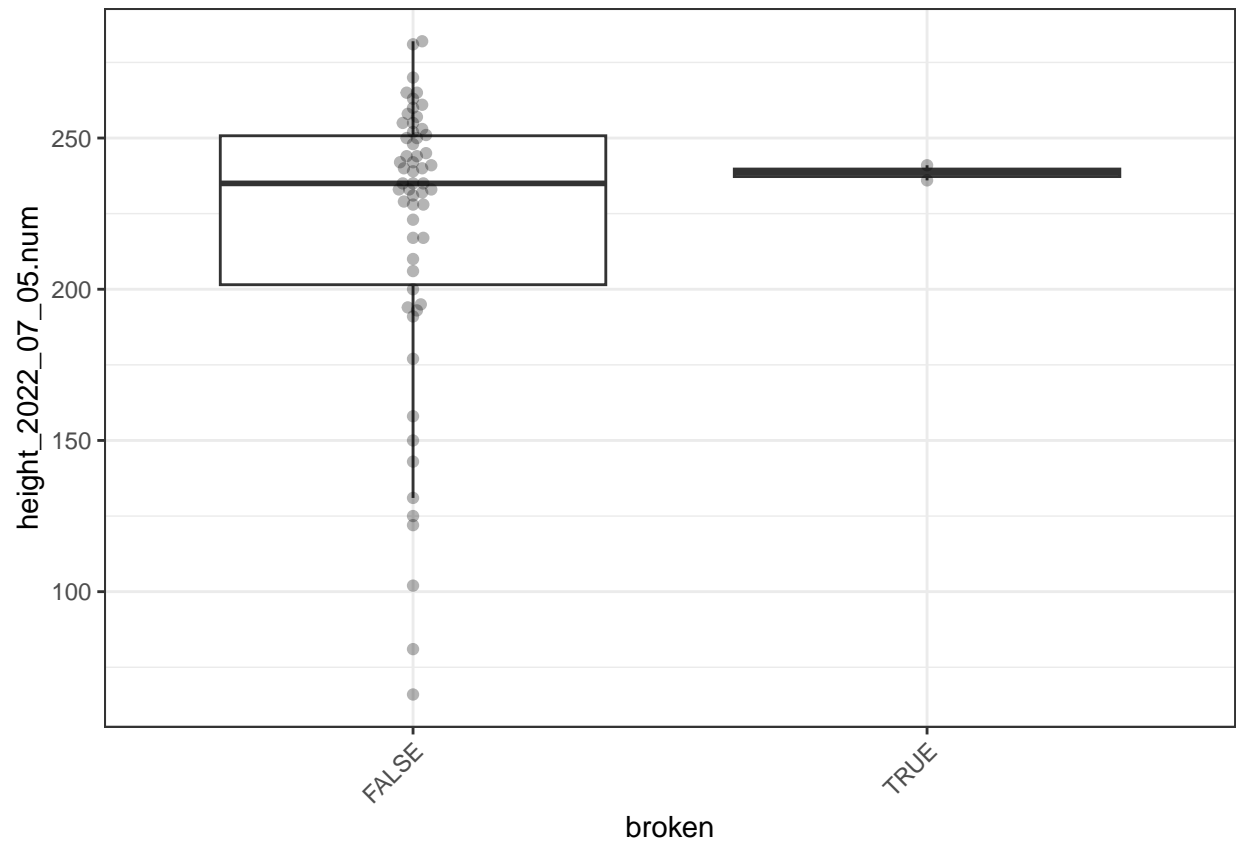
(m): Deleted 48 row(s) due to pairwise missing data.



6.2.6 We look at *broken*

```
f.gg.proto(type = "boxplot", y = `height_2022_07_05.num`, x = `broken`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

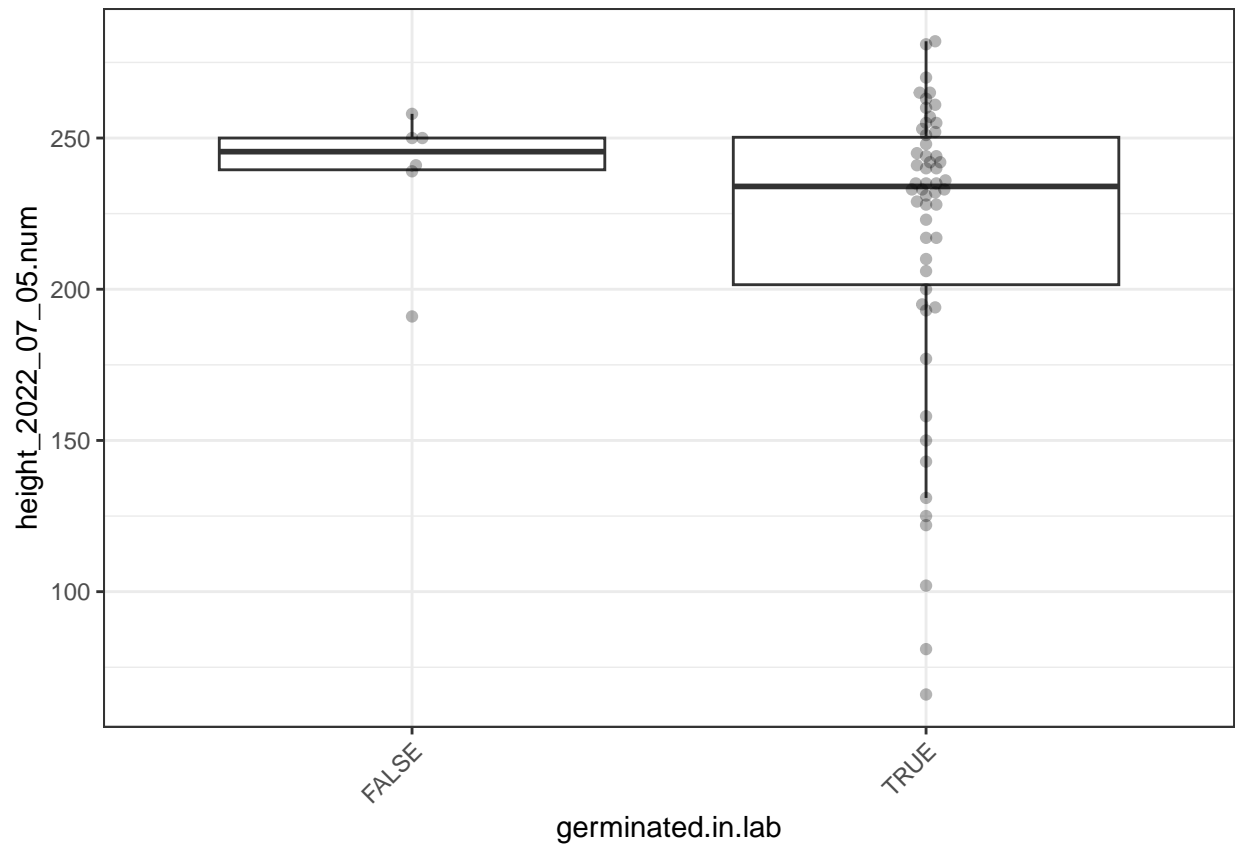
(m): Deleted 48 row(s) due to pairwise missing data.



6.2.7 We look at *germinated.in.lab*

```
f.gg.proto(type = "boxplot", y = `height_2022_07_05.num`, x = `germinated.in.lab`) +  
  scale_x_discrete(drop = FALSE) +  
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

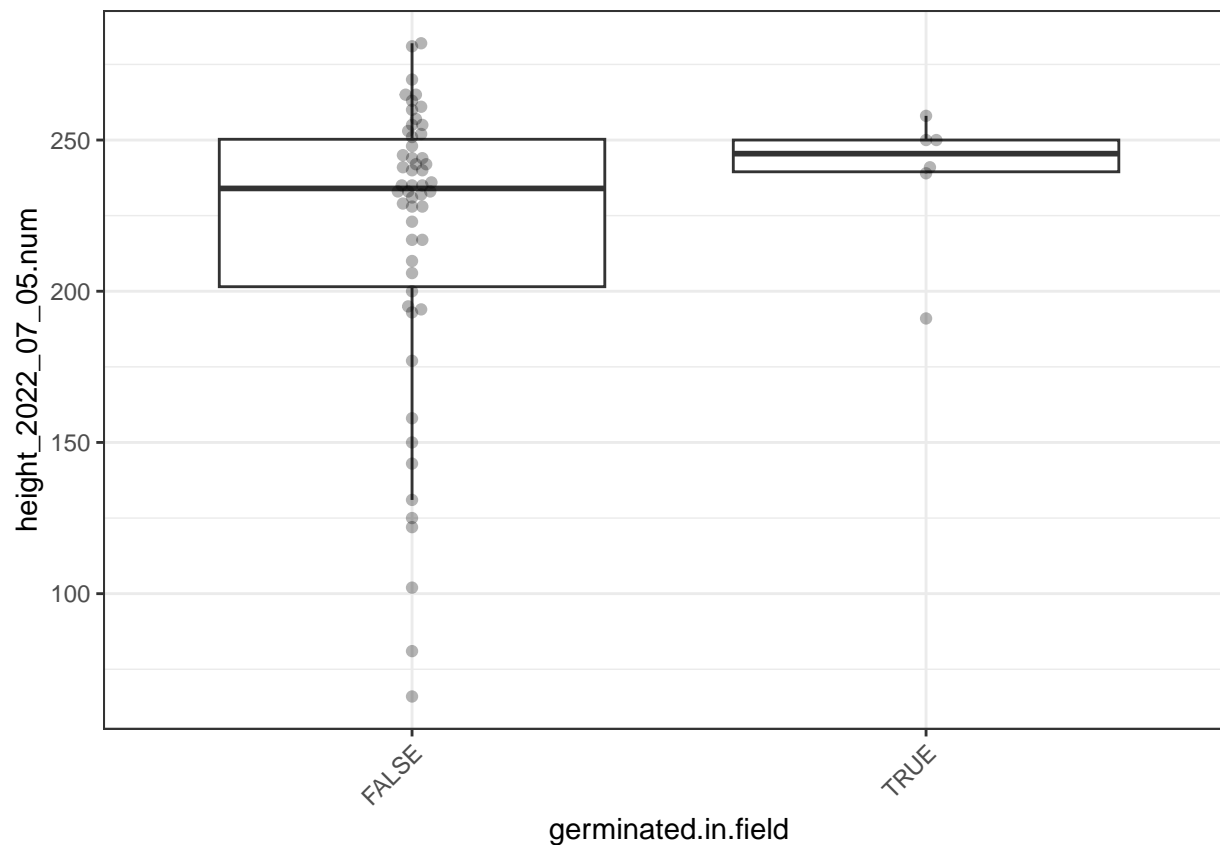
(m): Deleted 48 row(s) due to pairwise missing data.



6.2.8 We look at *germinated.in.field*

```
f.gg.proto(type = "boxplot", y = `height_2022_07_05.num`, x = `germinated.in.field`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

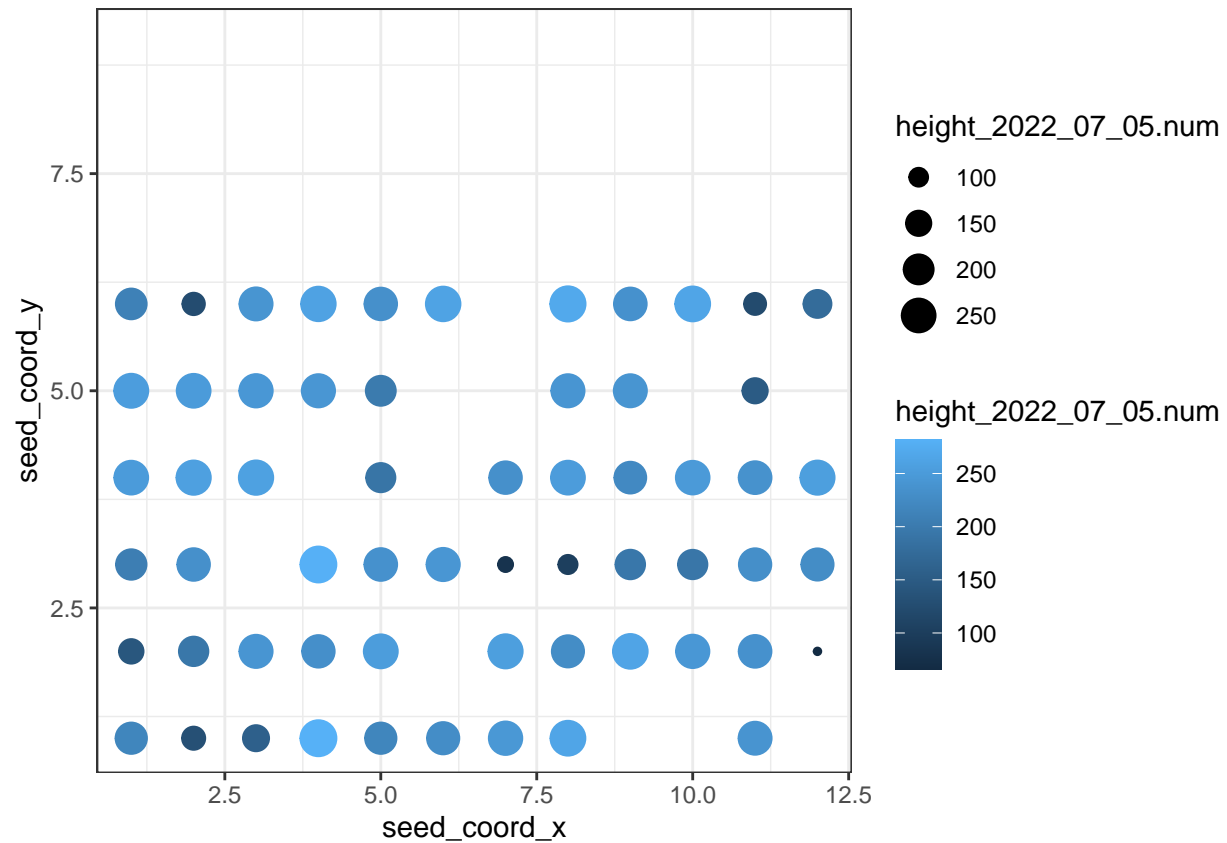
(m): Deleted 48 row(s) due to pairwise missing data.



6.2.9 We look at the coordinates

```
ggplot(d.maize, mapping = aes(x = seed_coord_x, y = seed_coord_y)) +
  geom_point(mapping = aes(size = height_2022_07_05.num,
                           colour = height_2022_07_05.num))
```

Warning: Removed 48 rows containing missing values (`geom_point()`).

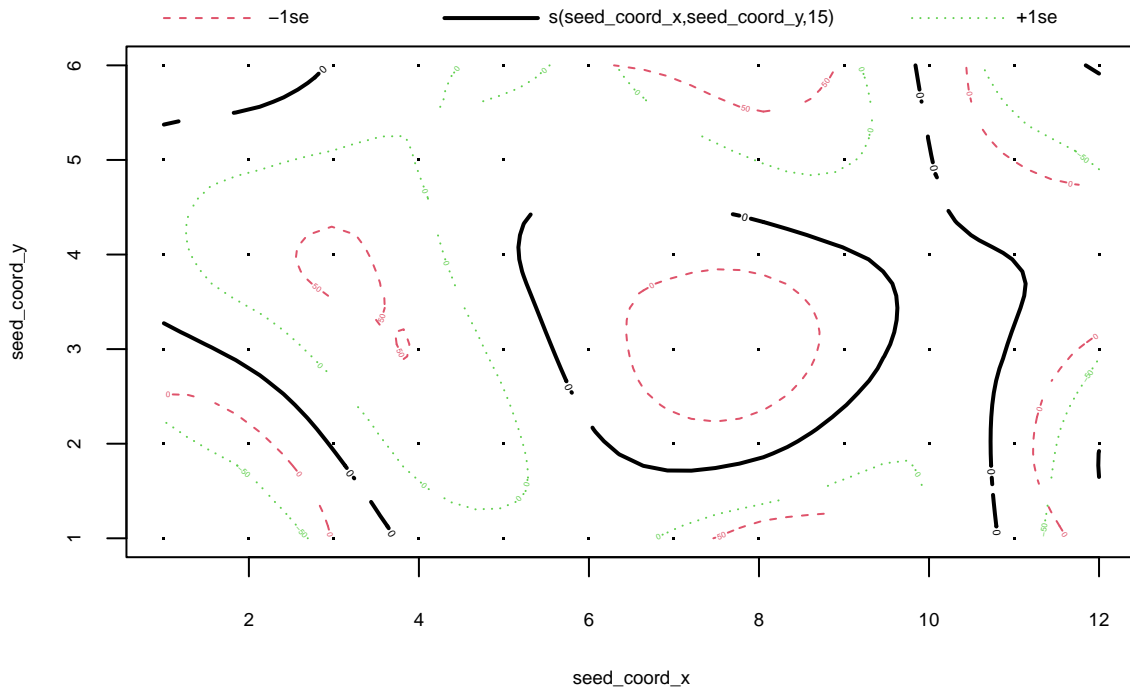


The bigger and lighter the points are, the higher the plant is.

There does not seem to be a relationship between the plant height and the lab position.

This is also evident in the plot below, which appears to display some unusual patterns but not a strong correlation between cob height and field position. Indeed, all the black lines represent a zero gradient, indicating the absence of any significant pattern.

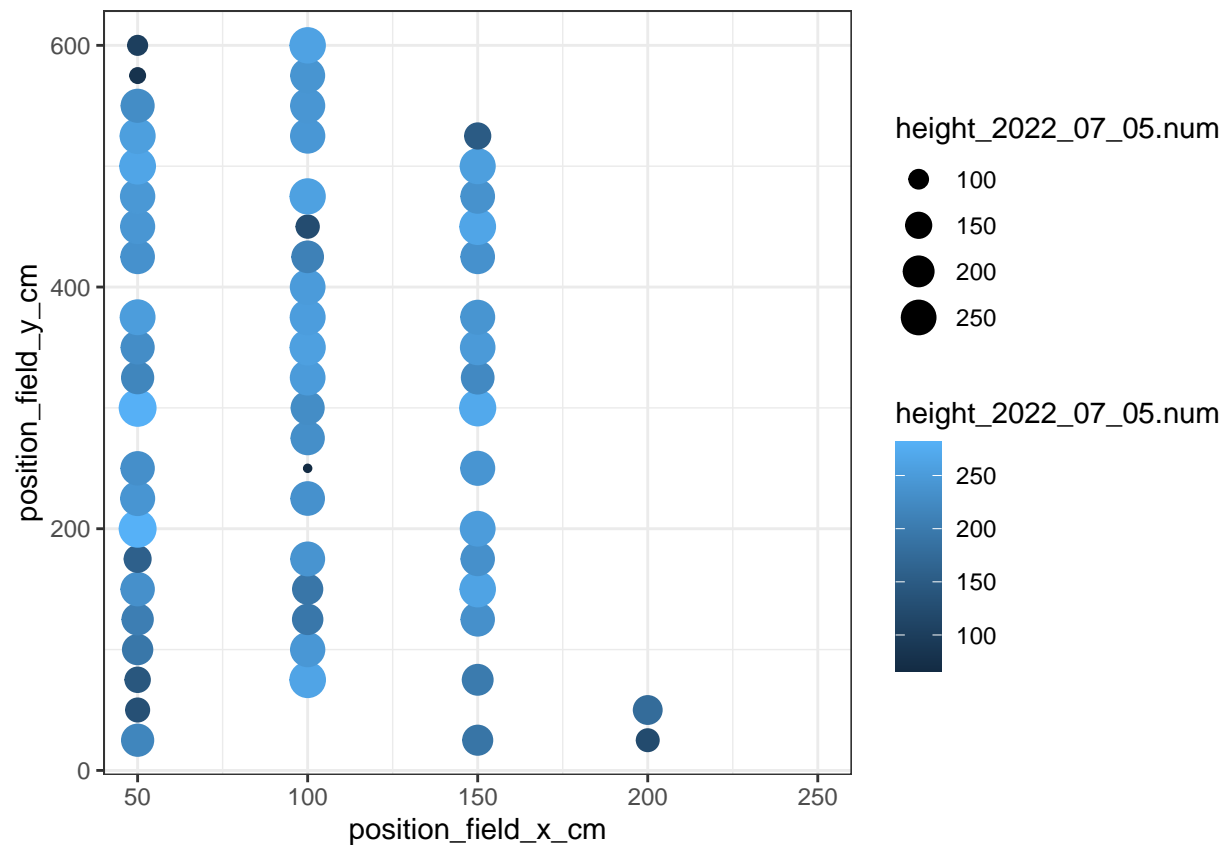
```
gam.height.coord <- gam(height_2022_07_05.num ~ s(seed_coord_x,
                                                    seed_coord_y,
                                                    k = 30),
                        data = d.maize)
plot(gam.height.coord)
```



6.3 We look at the position in the field

```
ggplot(d.maize, mapping = aes(x = position_field_x_cm, y = position_field_y_cm)) +
  geom_point(mapping = aes(size = height_2022_07_05.num,
    colour = height_2022_07_05.num))
```

Warning: Removed 48 rows containing missing values (`geom_point()`).

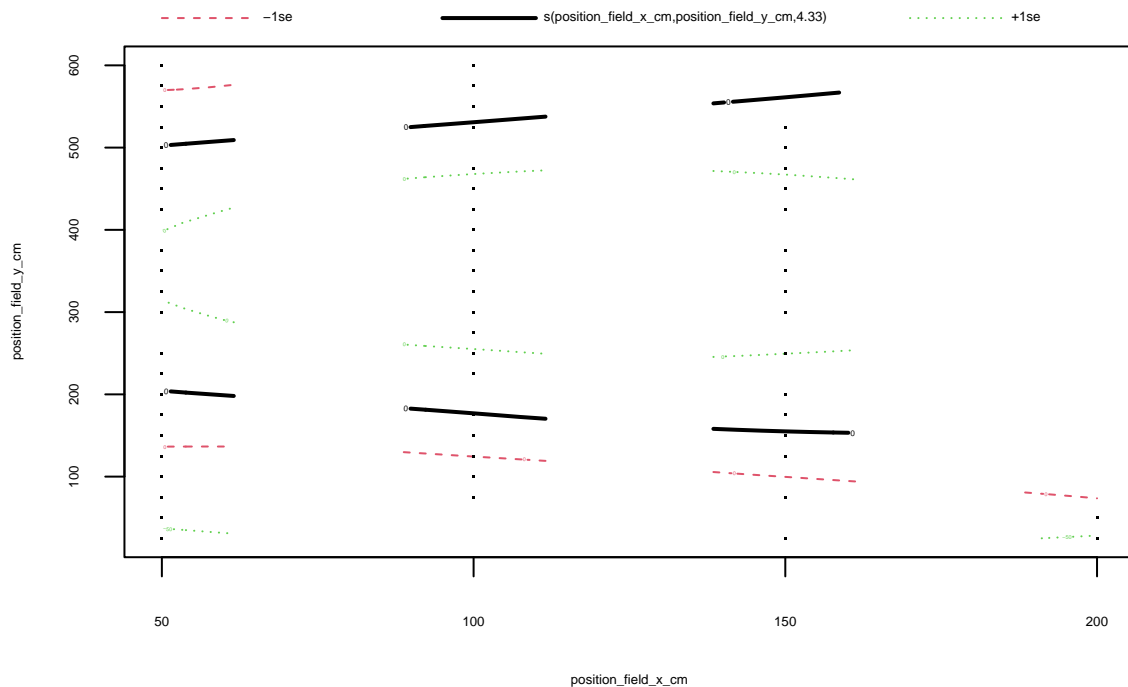


There appears to be a slight relationship between the plant height and the field position.

Specifically, plants at the extremities seem to be slightly smaller.

This observation is confirmed in the following plot.

```
gam.height.field <- gam(height_2022_07_05.num ~ s(position_field_x_cm,
                                                    position_field_y_cm,
                                                    k = 30),
                        data = d.maize)
plot(gam.height.field)
```



The pattern makes sense, since the plants that are surrounded by other plants, need to be higher to catch the sunlight.

6.4 Response variable: *germinated.yes*

6.4.1 We look at *pot.fac*

```
tab <- xtabs(~`germinated.yes` + `pot.fac`,
             addNA = TRUE,
             data = d.maize)
tab
```

	pot.fac																	
germinated.yes	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6	C1	C2	C3	C4	C5	C6
FALSE	0	1	1	0	1	0	0	0	1	2	1	1	3	2	2	2	0	1
TRUE	6	5	5	6	5	6	6	6	5	4	5	5	3	4	4	4	6	5

The next table converts the absolute numbers to the corresponding proportions [%] (rounded to 2 digits) with respect to *pot.fac*. I.e. the focus is on how *pot.fac* differs between categories of *germinated.yes*. Note that each row adds to 100%. Missing values are omitted.

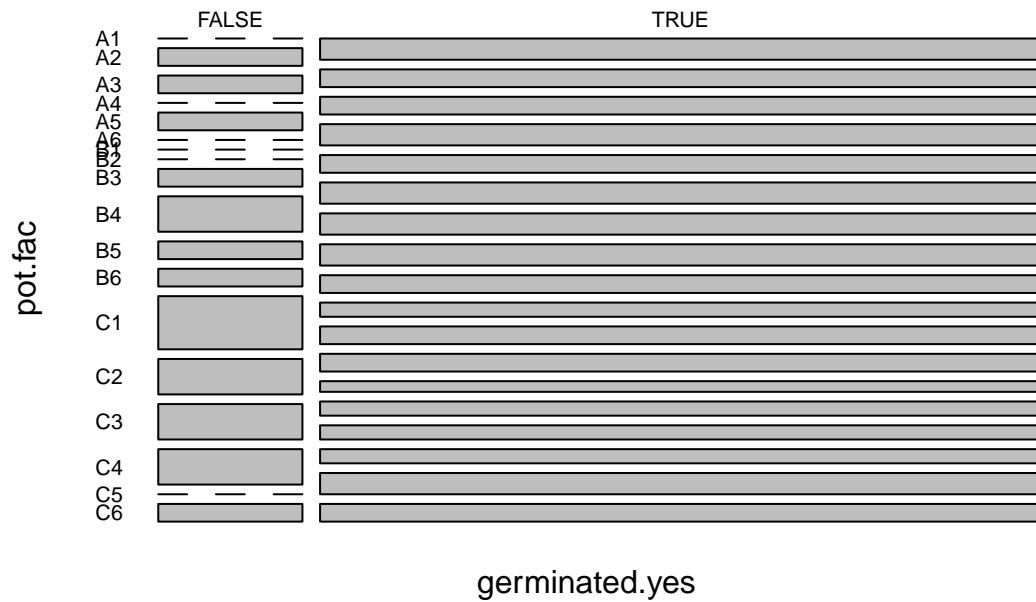
```
tab <- tab[!is.na(rownames(tab)), !is.na(colnames(tab))]
round(x = prop.table(x = tab, margin = 1) * 100, digits = 2)
```

	pot.fac												
germinated.yes	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6	C1
FALSE	0.0	5.6	5.6	0.0	5.6	0.0	0.0	0.0	5.6	11.1	5.6	5.6	16.7

	TRUE	6.7	5.6	5.6	6.7	5.6	6.7	6.7	6.7	5.6	4.4	5.6	5.6	3.3
	pot.fac													
germinated.yes	C2	C3	C4	C5	C6									
	FALSE	11.1	11.1	11.1	0.0	5.6								
	TRUE	4.4	4.4	4.4	6.7	5.6								

We plot the previous table

```
plot(x = tab, las = 1, main = "")
```



```
rm(tab)
```

6.4.2 We look at *soil.fac*

```
tab <- xtabs(~`germinated.yes` + `soil.fac`,
  addNA = TRUE,
  data = d.maize)
tab
```

	soil.fac				
germinated.yes	Bio	garden	Composana	herbs	mixture
	FALSE	4	5	5	4
	TRUE	20	19	31	20

The next table converts the absolute numbers to the corresponding proportions [%] (rounded to 2 digits) with respect to *soil.fac*. I.e. the focus is on how *soil.fac* differs between categories of *germinated.yes*. Note

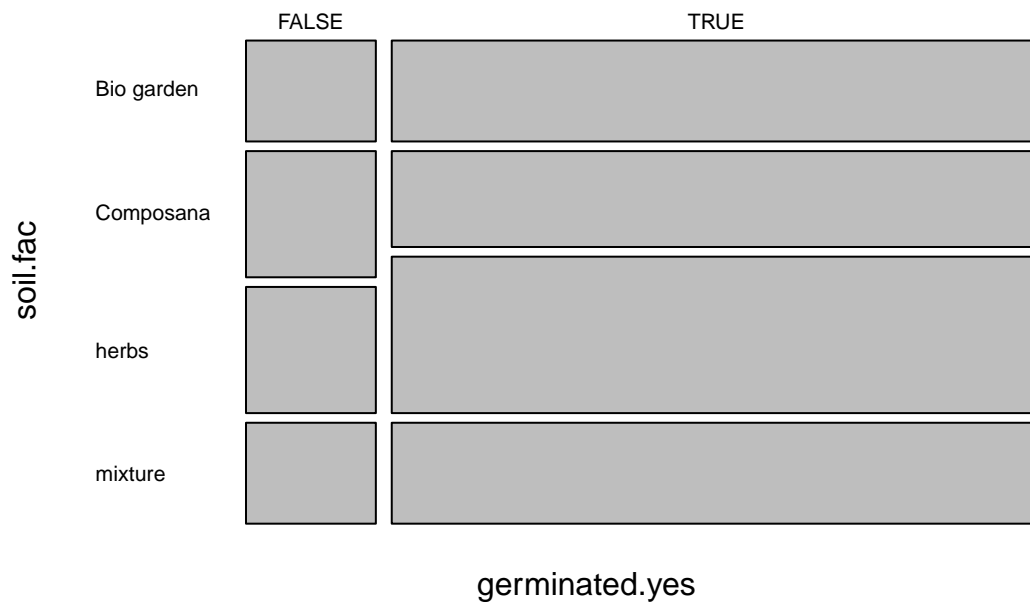
that each row adds to 100%. Missing values are omitted.

```
tab <- tab[!is.na(rownames(tab)), !is.na(colnames(tab))]  
round(x = prop.table(x = tab, margin = 1) * 100, digits = 2)
```

```
      soil.fac  
germinated.yes Bio garden Composana herbs mixture  
      FALSE      22      28      28      22  
      TRUE       22      21      34      22
```

We plot the previous table

```
plot(x = tab, las = 1, main = "")
```



```
rm(tab)
```

6.4.3 We look at *well.fac*

```
tab <- xtabs(~`germinated.yes` + `well.fac`,  
            addNA = TRUE,  
            data = d.maize)
```

```
tab
```

```
      well.fac  
germinated.yes a b c d e f  
      FALSE  3 3 2 6 3 1  
      TRUE  15 15 16 12 15 17
```

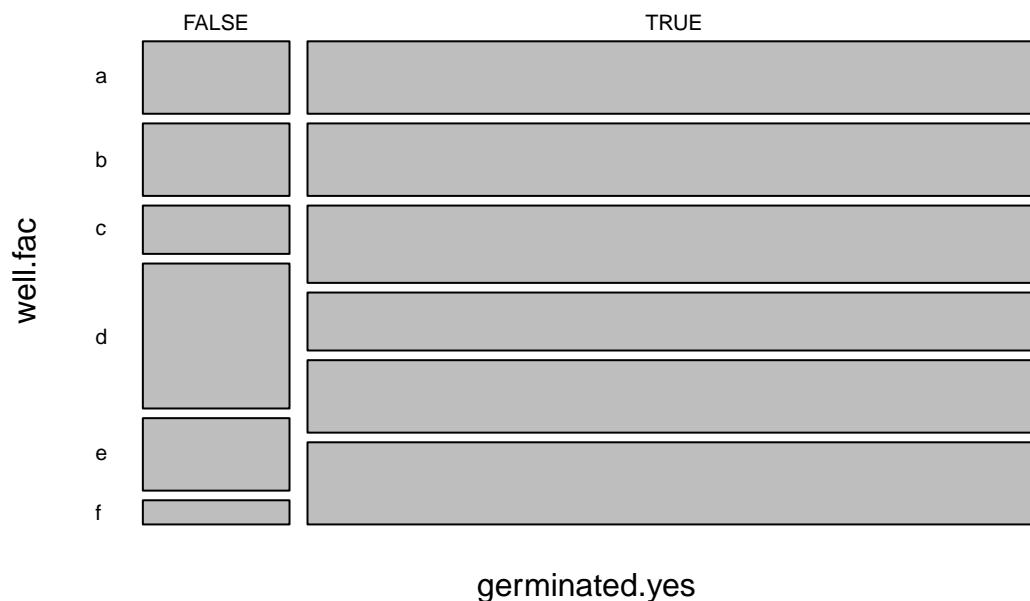
The next table converts the absolute numbers to the corresponding proportions [%] (rounded to 2 digits) with respect to *well.fac*. I.e. the focus is on how *well.fac* differs between categories of *germinated.yes*. Note that each row adds to 100%. Missing values are omitted.

```
tab <- tab[!is.na(rownames(tab)), !is.na(colnames(tab))]  
round(x = prop.table(x = tab, margin = 1) * 100, digits = 2)
```

```
      well.fac  
germinated.yes  a    b    c    d    e    f  
      FALSE 16.7 16.7 11.1 33.3 16.7  5.6  
      TRUE  16.7 16.7 17.8 13.3 16.7 18.9
```

We plot the previous table

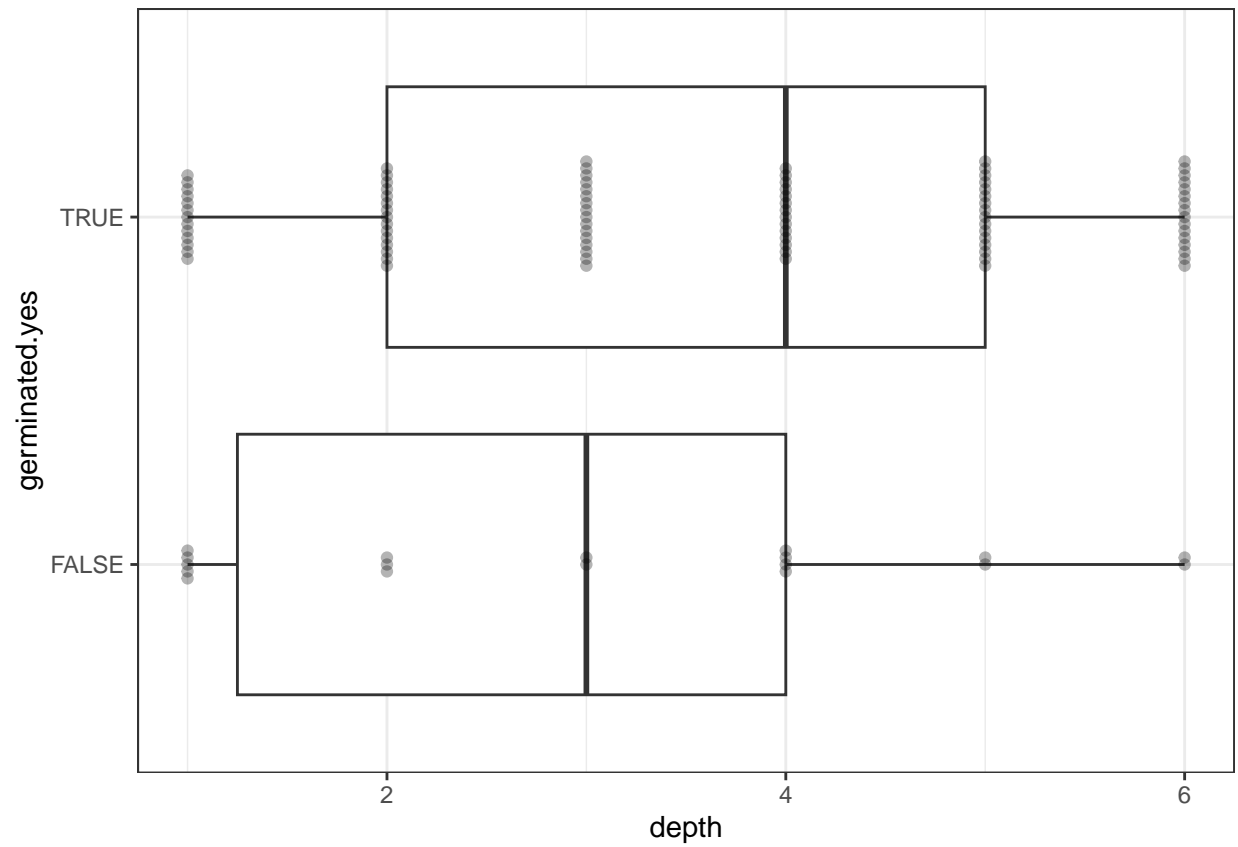
```
plot(x = tab, las = 1, main = "")
```



```
rm(tab)
```

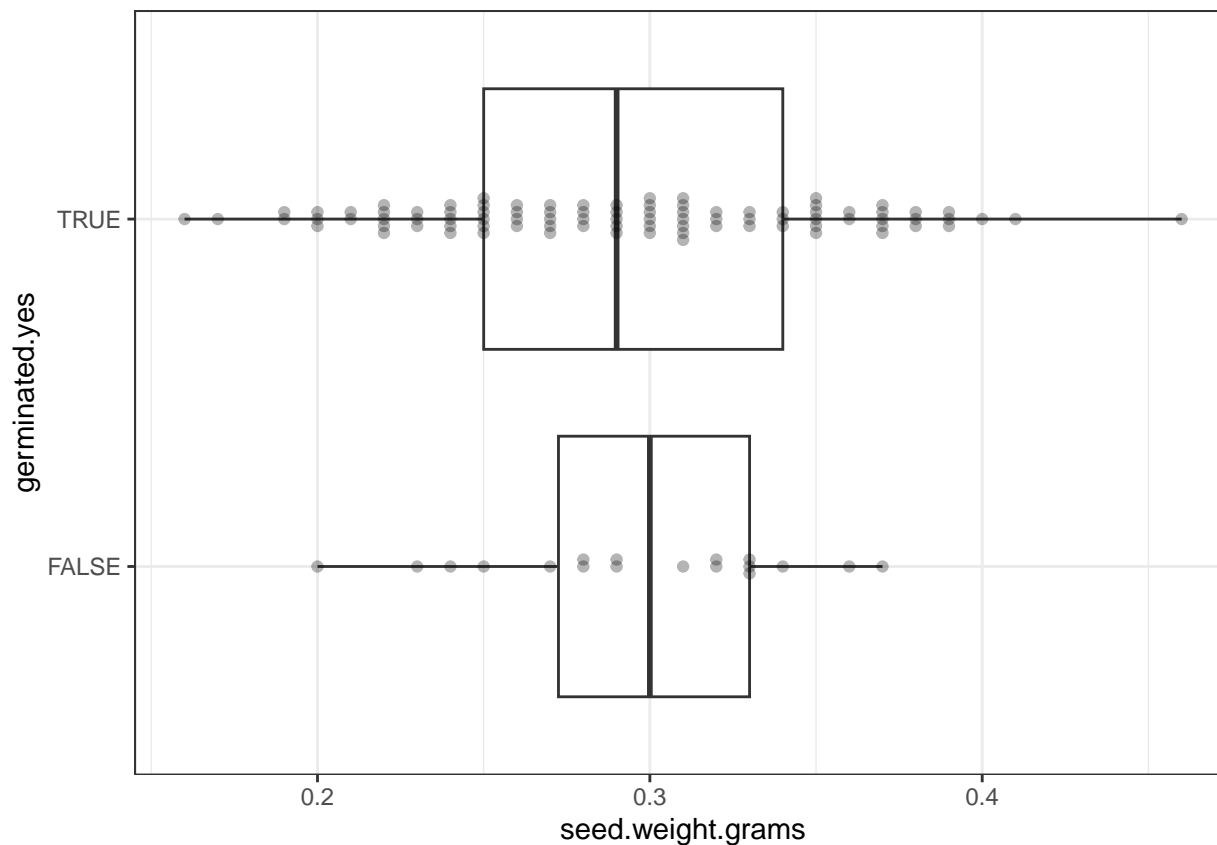
6.4.4 We look at *depth*

```
f.gg.proto(type = "boxplot", y = `germinated.yes`, x = `depth`) +  
  scale_y_discrete(drop = FALSE)
```



6.4.5 We look at *seed.weight.grams*

```
f.gg.proto(type = "boxplot", y = `germinated.yes`, x = `seed.weight.grams`) +  
  scale_y_discrete(drop = FALSE)
```



6.4.6 We look at *broken*

```
tab <- xtabs(~`germinated.yes` + `broken`,
             addNA = TRUE,
             data = d.maize)
tab
```

	broken	
germinated.yes	FALSE	TRUE
FALSE	17	1
TRUE	87	3

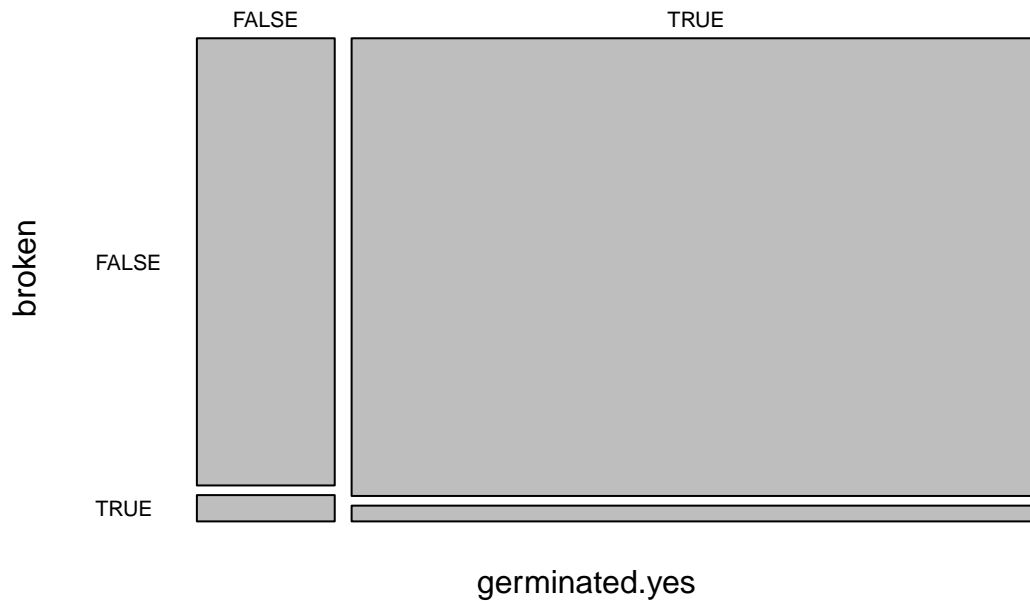
The next table converts the absolute numbers to the corresponding proportions [%] (rounded to 2 digits) with respect to *broken*. I.e. the focus is on how *broken* differs between categories of *germinated.yes*. Note that each row adds to 100%. Missing values are omitted.

```
tab <- tab[!is.na(rownames(tab)), !is.na(colnames(tab))]
round(x = prop.table(x = tab, margin = 1) * 100, digits = 2)
```

	broken	
germinated.yes	FALSE	TRUE
FALSE	94.4	5.6
TRUE	96.7	3.3

We plot the previous table

```
plot(x = tab, las = 1, main = "")
```



```
rm(tab)
```

6.4.7 We look at *germinated.in.lab*

```
tab <- xtabs(~`germinated.yes` + `germinated.in.lab`,
            addNA = TRUE,
            data = d.maize)
tab
```

```

      germinated.in.lab
germinated.yes FALSE TRUE
      FALSE     18     0
      TRUE      6    84
```

The next table converts the absolute numbers to the corresponding proportions [%] (rounded to 2 digits) with respect to *germinated.in.lab*. I.e. the focus is on how *germinated.in.lab* differs between categories of *germinated.yes*. Note that each row adds to 100%. Missing values are omitted.

```
tab <- tab[!is.na(rownames(tab)), !is.na(colnames(tab))]
round(x = prop.table(x = tab, margin = 1) * 100, digits = 2)
```

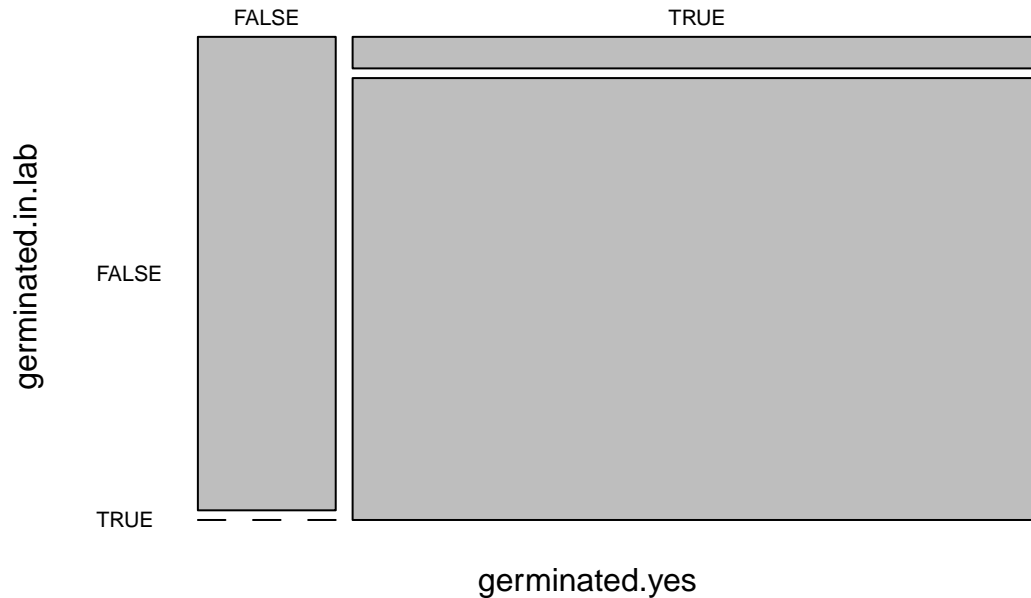
```

      germinated.in.lab
germinated.yes FALSE  TRUE
```

```
FALSE 100.0  0.0
TRUE   6.7  93.3
```

We plot the previous table

```
plot(x = tab, las = 1, main = "")
```



```
rm(tab)
```

6.4.8 We look at *germinated.in.field*

```
tab <- xtabs(~`germinated.yes` + `germinated.in.field`,
             addNA = TRUE,
             data = d.maize)
tab
```

```
             germinated.in.field
germinated.yes FALSE TRUE
FALSE         18    0
TRUE          84    6
```

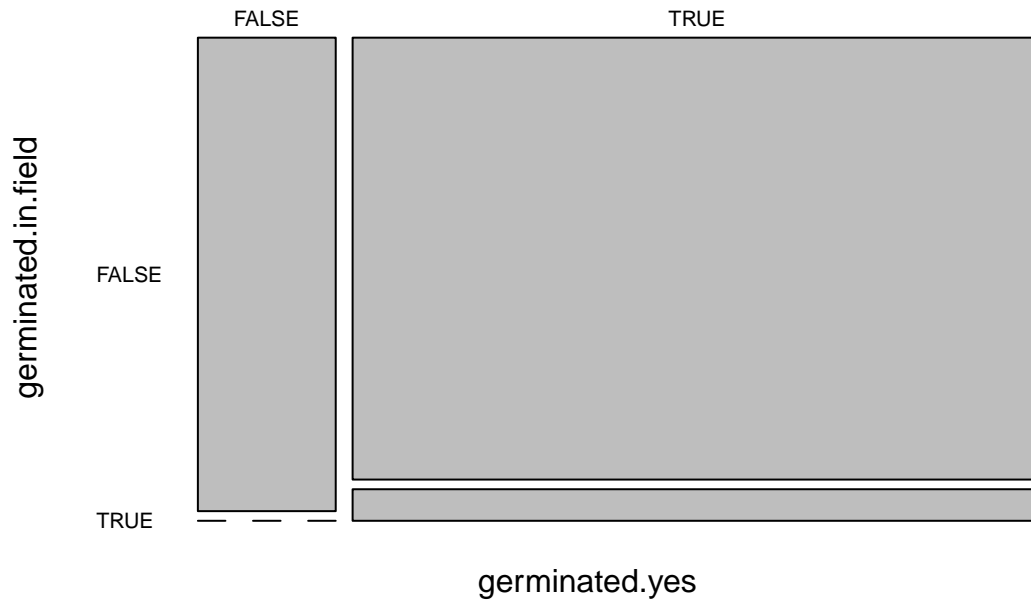
The next table converts the absolute numbers to the corresponding proportions [%] (rounded to 2 digits) with respect to *germinated.in.field*. I.e. the focus is on how *germinated.in.field* differs between categories of *germinated.yes*. Note that each row adds to 100%. Missing values are omitted.

```
tab <- tab[!is.na(rownames(tab)), !is.na(colnames(tab))]  
round(x = prop.table(x = tab, margin = 1) * 100, digits = 2)
```

```
      germinated.in.field  
germinated.yes FALSE  TRUE  
      FALSE 100.0   0.0  
      TRUE   93.3   6.7
```

We plot the previous table

```
plot(x = tab, las = 1, main = "")
```

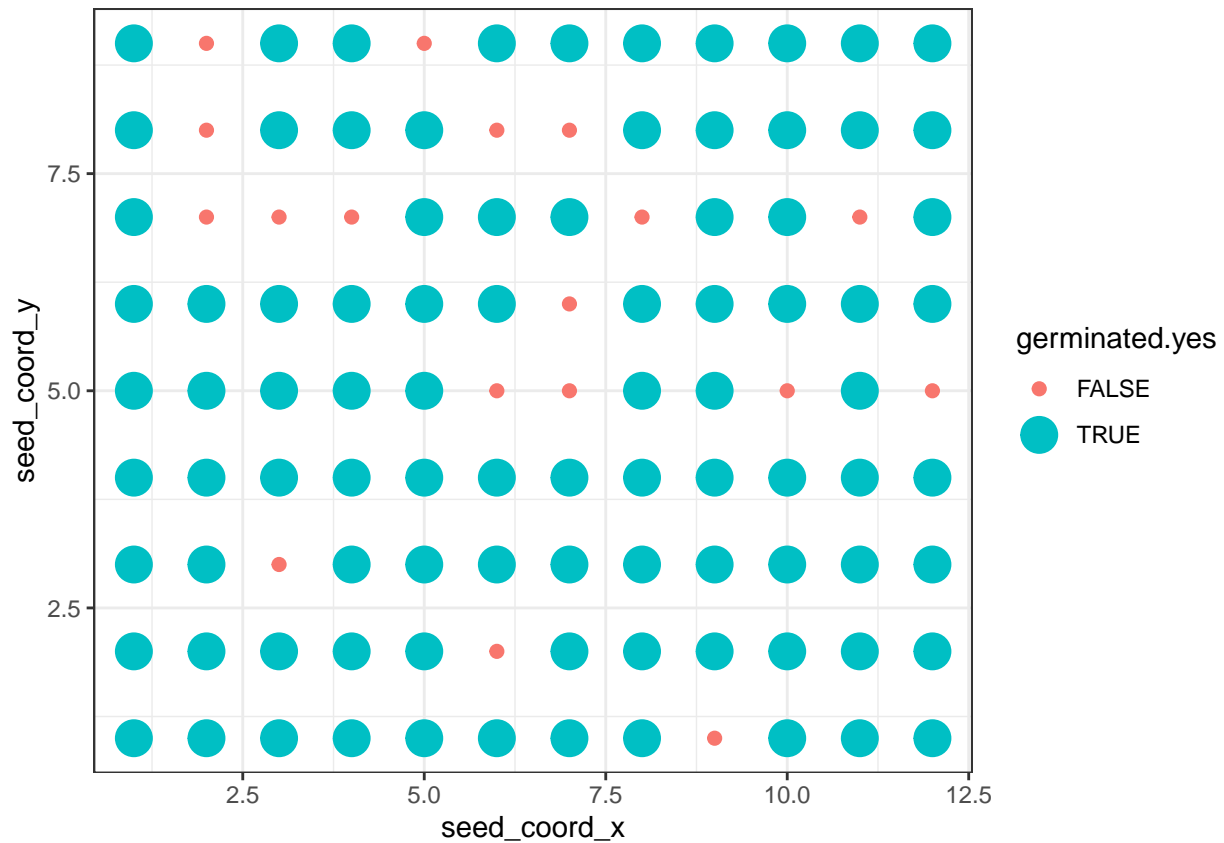


```
rm(tab)
```

6.4.9 We look at the coordinates

```
ggplot(d.maize, mapping = aes(x = seed_coord_x, y = seed_coord_y)) +  
  geom_point(mapping = aes(size = germinated.yes,  
                           colour = germinated.yes))
```

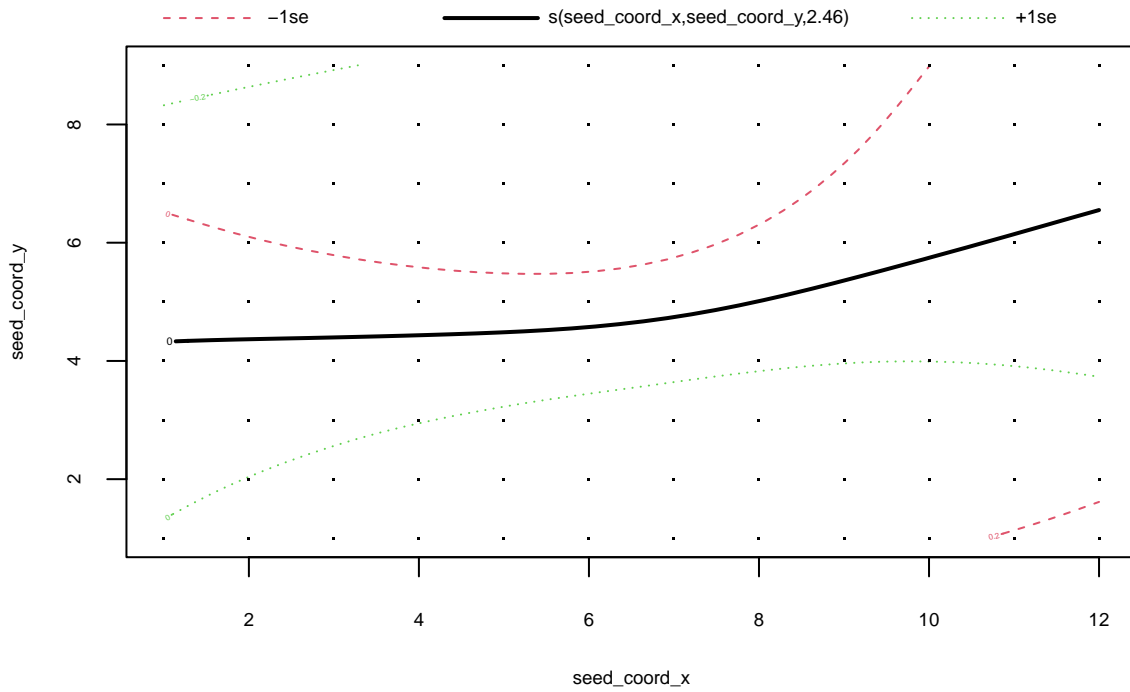
Warning: Using size for a discrete variable is not advised.



It seems that the seeds positioned on the upper half, struggled a little bit more than the others to germinate. This pattern is also highlighted in the following plot.

The response variable *germinated.yes* does not tell us whether the seed germinated in the lab or in the field, thus these plots are of small relevance.

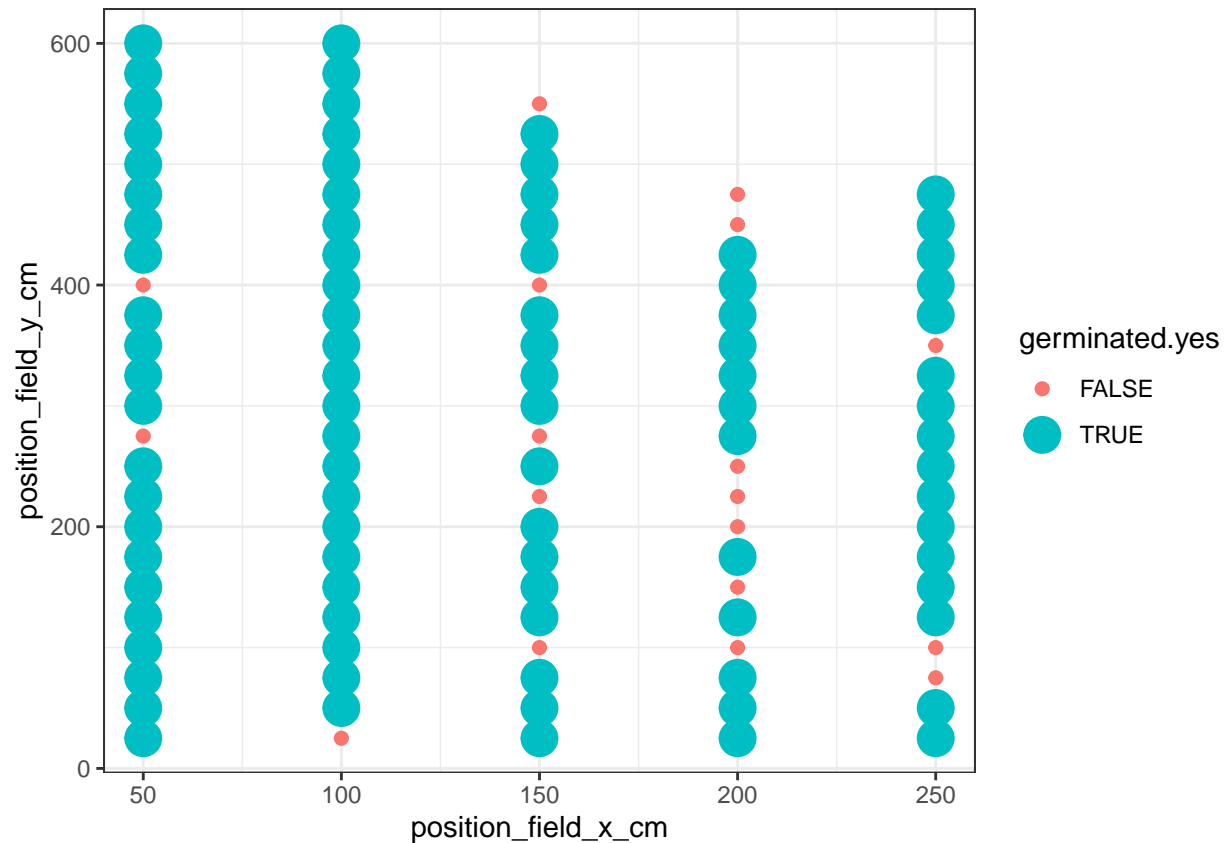
```
gam.germinated.yes.coord <- gam(germinated.yes ~ s(seed_coord_x,
                                                    seed_coord_y,
                                                    k = 30),
                                data = d.maize)
plot(gam.germinated.yes.coord)
```



6.5 We look at the position in the field

```
ggplot(d.maize, mapping = aes(x = position_field_x_cm, y = position_field_y_cm)) +
  geom_point(mapping = aes(size = germinated.yes,
                           colour = germinated.yes))
```

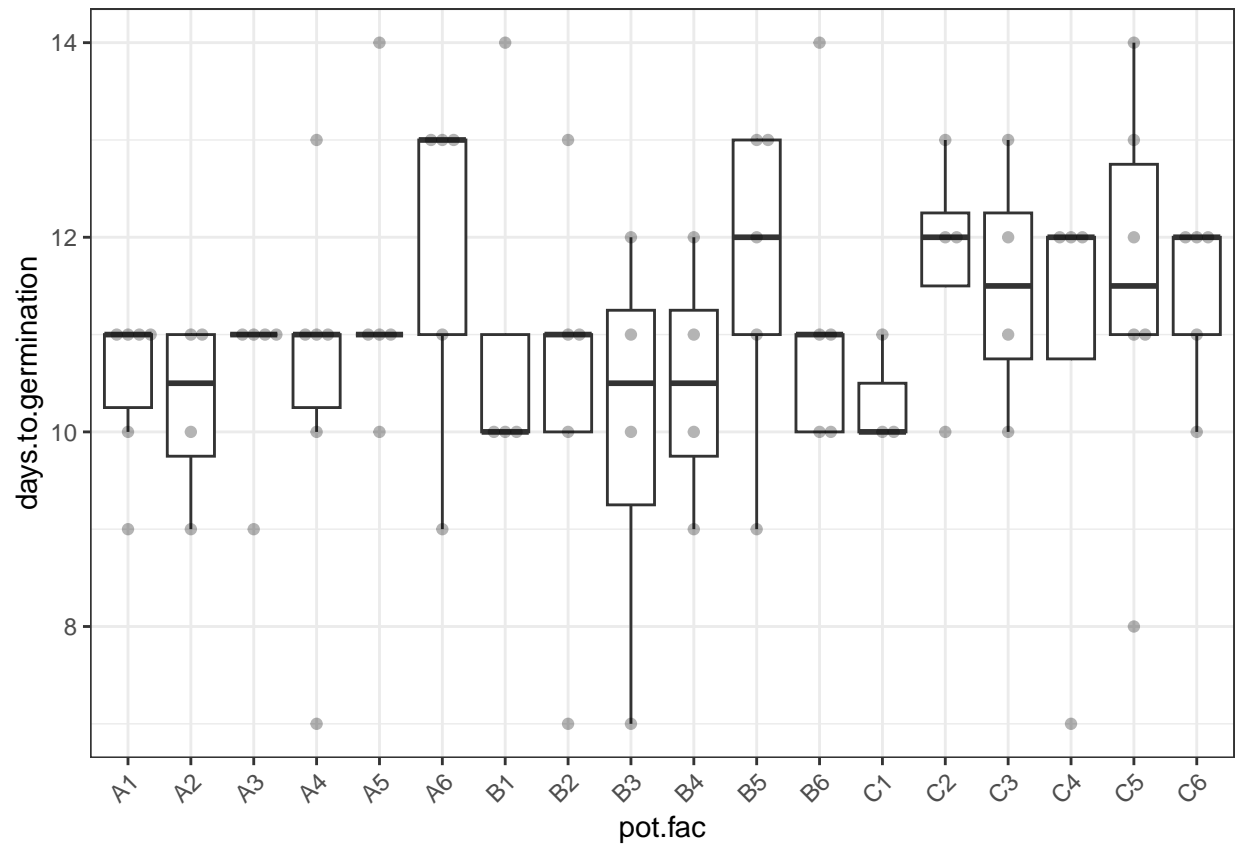
Warning: Using size for a discrete variable is not advised.



The seeds that did not germinated are positioned on the right half of the field. This is confirmed in the graph below.

The response variable *germinated.yes* does not tell us whether the seed germinated in the lab or in the field, thus these plots are of small relevance.

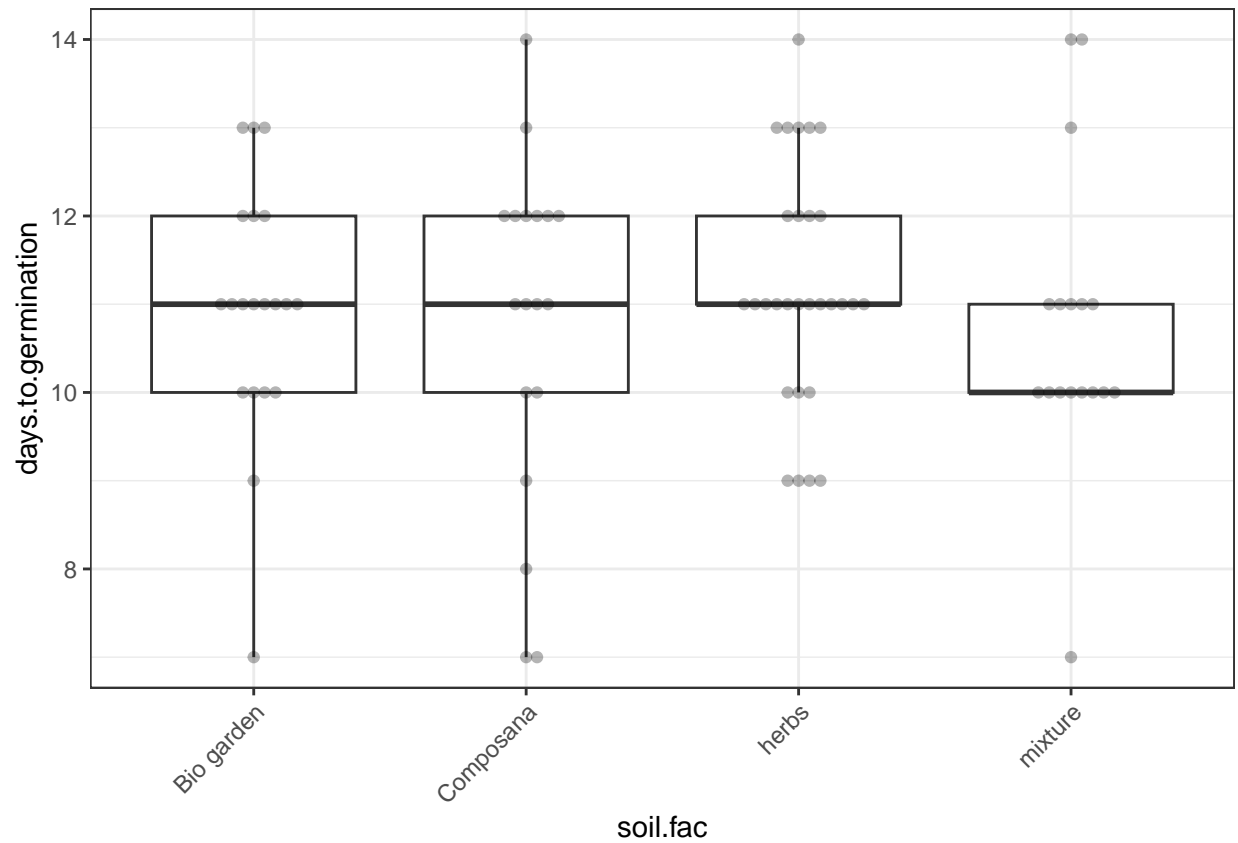
```
gam.germinated.yes.field <- gam(germinated.yes ~ s(position_field_x_cm,
                                                    position_field_y_cm,
                                                    k = 30),
                                data = d.maize)
plot(gam.germinated.yes.field)
```

6.6.2 We look at *soil.fac*

```
f.gg.proto(type = "boxplot", y = `days.to.germination`, x = `soil.fac`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

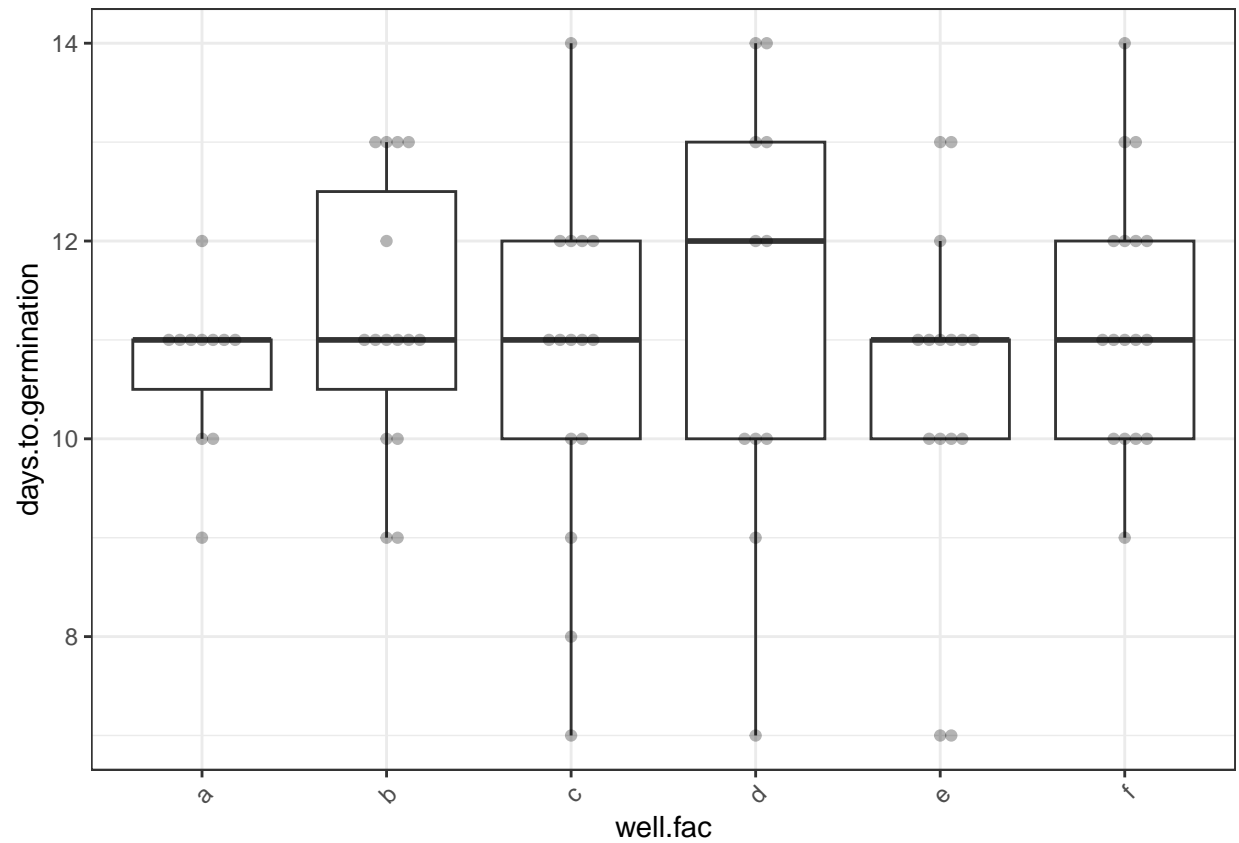
(m): Deleted 24 row(s) due to pairwise missing data.



6.6.3 We look at *well.fac*

```
f.gg.proto(type = "boxplot", y = `days.to.germination`, x = `well.fac`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

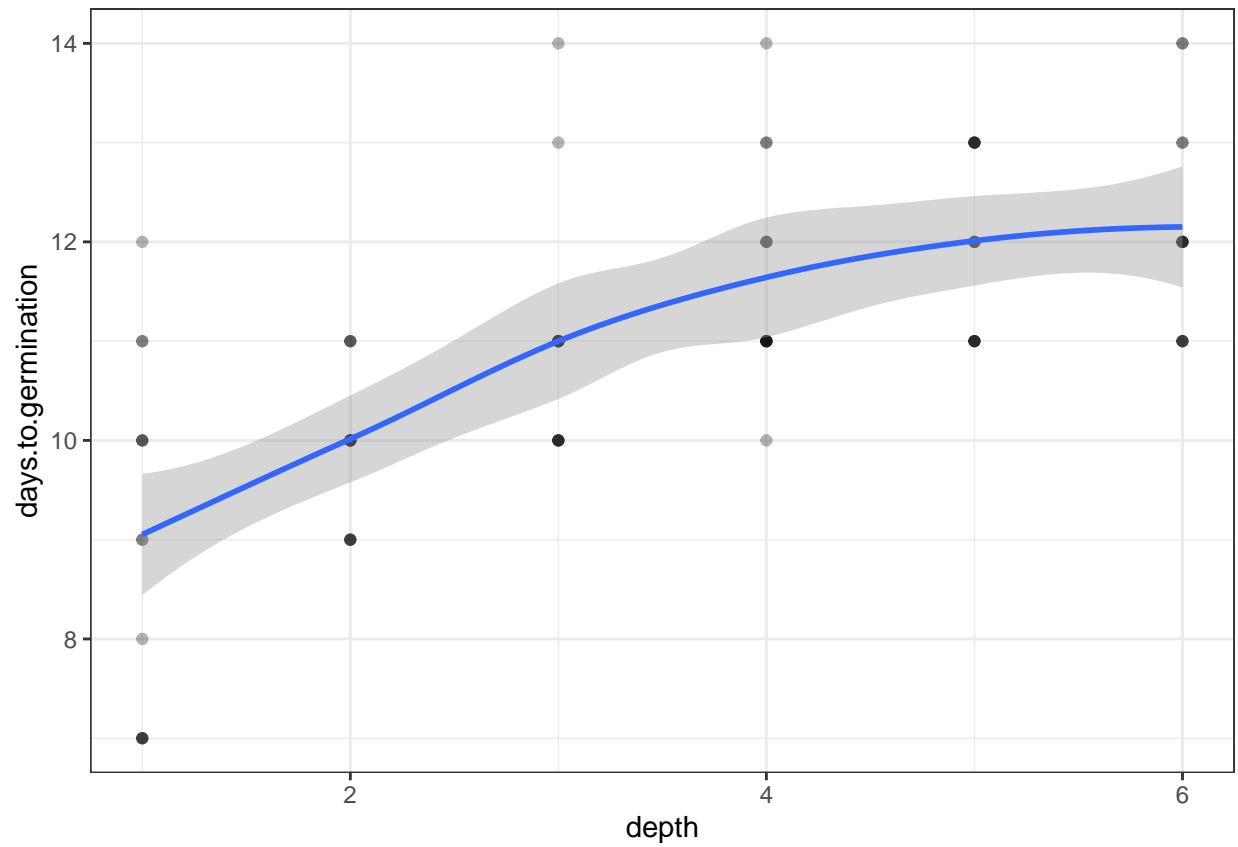
(m): Deleted 24 row(s) due to pairwise missing data.



6.6.4 We look at *seed.weight.grams*

```
f.gg.proto(type = "scatter", y = `days.to.germination`, x = `depth`)
```

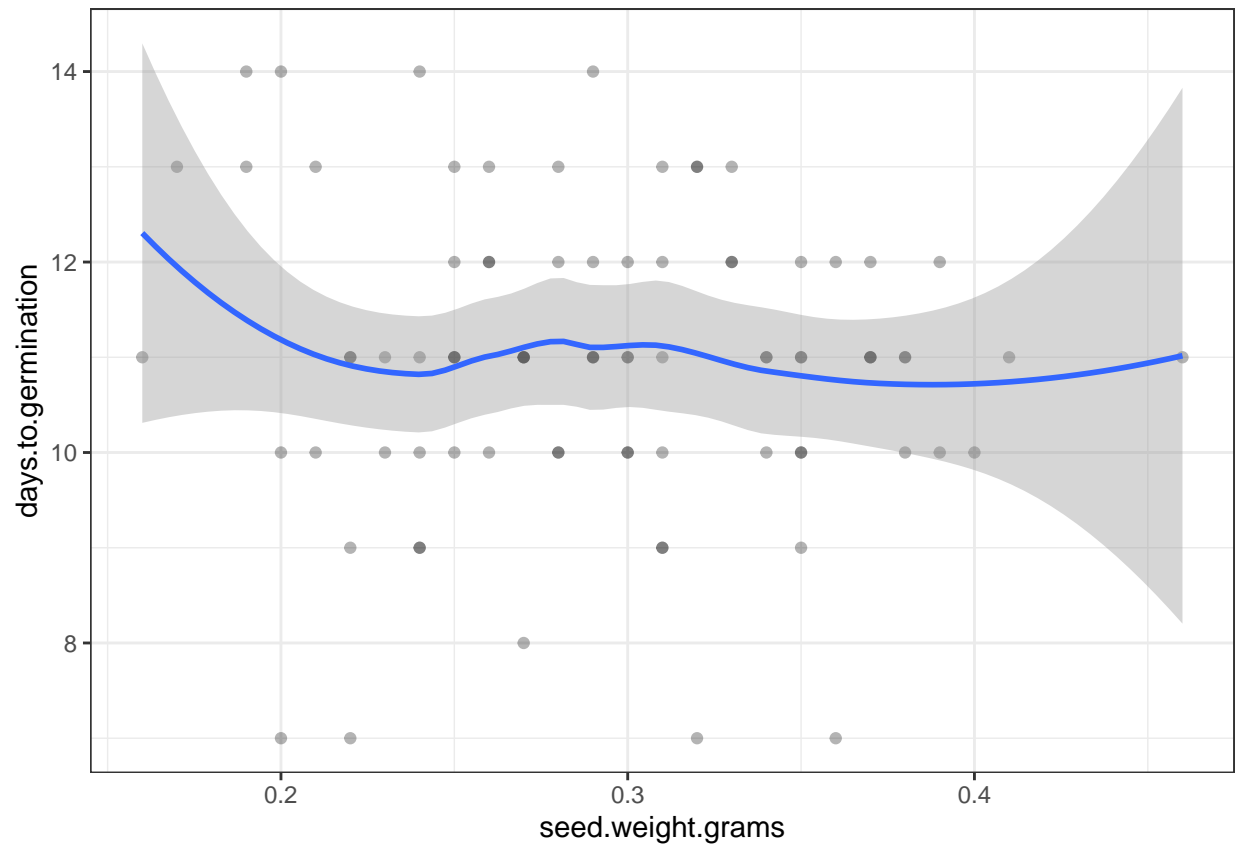
(m): Deleted 24 row(s) due to pairwise missing data.



6.6.5 We look at *depth*

```
f.gg.proto(type = "scatter", y = `days.to.germination`, x = `seed.weight.grams`)
```

(m): Deleted 24 row(s) due to pairwise missing data.

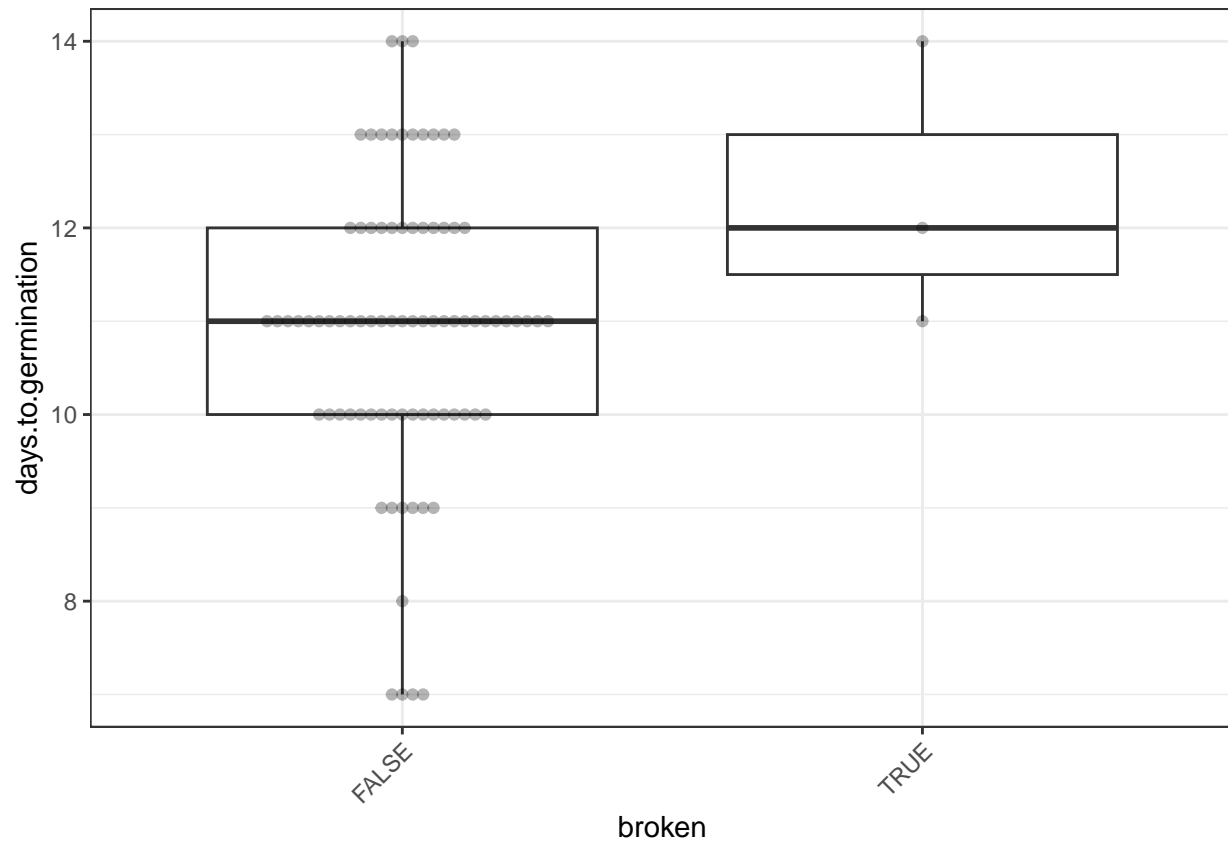


We can observe a positive relationship between the two variables. This is expected because the deeper the seed is in the soil, the longer it takes to emerge from it.

6.6.6 We look at *broken*

```
f.gg.proto(type = "boxplot", y = `days.to.germination`, x = `broken`) +
  scale_x_discrete(drop = FALSE) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

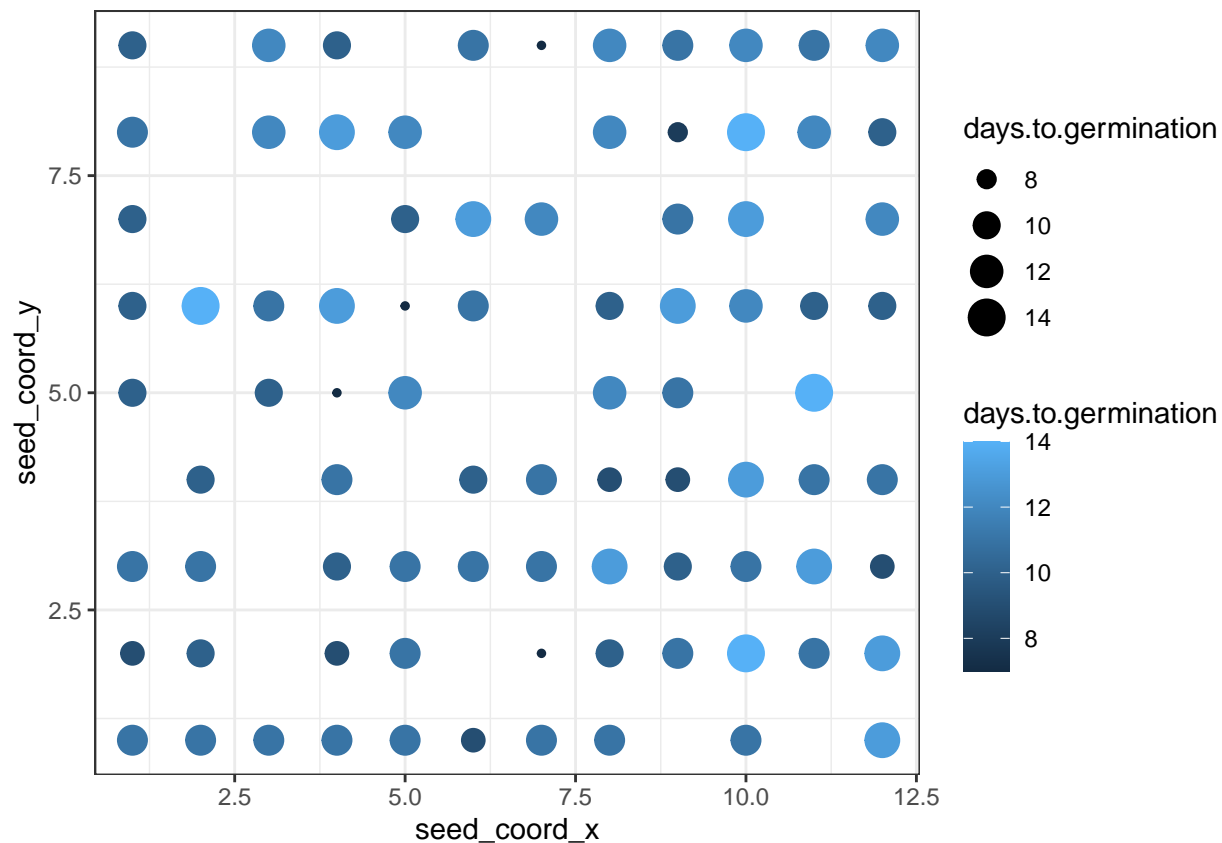
(m): Deleted 24 row(s) due to pairwise missing data.



6.6.7 We look at the coordinates

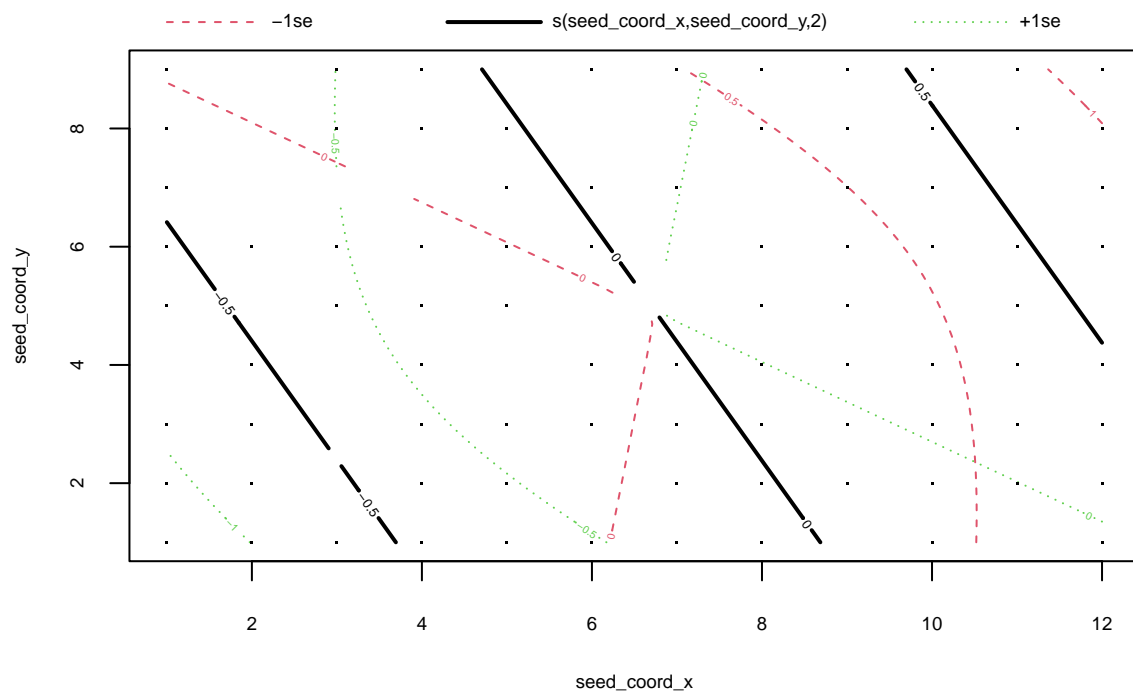
```
ggplot(d.maize, mapping = aes(x = seed_coord_x, y = seed_coord_y)) +  
  geom_point(mapping = aes(size = days.to.germination,  
                           colour = days.to.germination))
```

Warning: Removed 24 rows containing missing values (`geom_point()`).



The plot appears to illustrate a relationship between *days to germination* and the position within the lab. Seeds in the upper-right maizeer of the plot seem to germinate in fewer days compared to those in the lower-left maizeer. This observation aligns with the trend evident in the plot below, where we observe a higher gradient in the upper-right maizeer and a lower one in the lower-left maizeer.

```
gam.days.to.germination.coord <- gam(days.to.germination ~ s(seed_coord_x,
  seed_coord_y,
  k = 30),
  data = d.maize)
plot(gam.days.to.germination.coord)
```



7 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] survminer_0.4.9  ggpubr_0.6.0    survival_3.5-5  mgcv_1.8-42
[5] nlme_3.1-162     tidyr_1.3.0     tableone_0.13.2 ggbeeswarm_0.7.2
[9] stringr_1.5.0    lubridate_1.9.3 ggplot2_3.4.4   kableExtra_1.3.4
[13] dplyr_1.1.3      checkpoint_1.0.2 knitr_1.44
```

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

```
[1] gtable_0.3.4      beeswarm_0.4.0   xfun_0.40        rstatix_0.7.2
[5] lattice_0.21-8    vctrs_0.6.4      tools_4.3.1      generics_0.1.3
[9] tibble_3.2.1      fansi_1.0.5      pkgconfig_2.0.3  Matrix_1.6-1.1
[13] data.table_1.14.8 webshot_0.5.5    lifecycle_1.0.3  farver_2.1.1
[17] compiler_4.3.1    munsell_0.5.0    mitools_2.4      carData_3.0-5
[21] survey_4.2-1      vipor_0.4.5      htmltools_0.5.6.1 yaml_2.3.7
[25] crayon_1.5.2      pillar_1.9.0     car_3.1-2        abind_1.4-5
[29] km.ci_0.5-6       tidyselect_1.2.0 rvest_1.0.3      digest_0.6.33
[33] stringi_1.7.12    purrr_1.0.2      labeling_0.4.3   forcats_1.0.0
[37] splines_4.3.1     labelled_2.12.0  fastmap_1.1.1    grid_4.3.1
[41] colorspace_2.1-0 cli_3.6.1        magrittr_2.0.3   utf8_1.2.4
[45] broom_1.0.5       withr_2.5.1      scales_1.2.1     backports_1.4.1
[49] timechange_0.2.0 rmarkdown_2.25   httr_1.4.7       gridExtra_2.3
[53] ggsignif_0.6.4    hms_1.1.3        zoo_1.8-12       evaluate_0.22
[57] haven_2.5.3       KMsurv_0.1-5     viridisLite_0.4.2 survMisc_0.5.6
[61] rlang_1.1.1       xtable_1.8-4     glue_1.6.2       DBI_1.1.3
[65] xml2_1.3.5        svglite_2.1.2    rstudioapi_0.15.0 R6_2.5.1
[69] systemfonts_1.0.5
```