

les 6-5

2024-05-06

Onderzoeksopzet

Plan van aanpak

In dit plan van aanpak wordt globaal beschreven hoe dit experiment wordt aangepakt, er wordt verwezen naar protocollen die als bijlage bijgevoegd zijn. Alle documenten die benodigd zijn om dingen in te voeren zijn ook bijgevoegd.

Experiment

In dit experiment wordt de invloed van verschillende zoutconcentraties op de groei van tuinkers en alfalfa onderzocht. Er is een controlegroep met 0% zout, een groep met 0.03% zout, 0.06% zout en 0.09% zout, er wordt gedemineraliseerd water gebruikt om de zoutconcentratie zo precies mogelijk te kunnen toevoegen. Alle groepen worden in duplo uitgevoerd. De plantjes worden allemaal op 1 locaties gezaaid en geoogst.

Onderzoeksvraag

Wat is de invloed van verschillende zoutconcentraties op de groei van tuinkers en alfalfa?

Nulhypothese

Er is geen invloed van verschillende zoutconcentraties op de groei van tuinkers en alfalfa.

Hypothese

Een oplopende zoutconcentratie hoger dan 0.03% zal een negatieve invloed hebben op de groei van tuinkers en alfalfa.

Protocollen

Verwijzing naar protocollen... komt nog

Theoretische achtergrond

De concentratie van opgeloste stoffen in water, welke in de omgeving van een plant is, kan een grote invloed hebben op de plant en de groei van deze plant. In een hypotonische omgeving, de omgeving heeft dan een lagere concentratie opgeloste stoffen, zal het water van de omgeving in de cel van de plant gaan. De cel zal onder meer druk komen te staan, en opzwellen, wanneer deze situatie voor langere tijd niet veranderd zal de cel knappen. In een hypertonische situatie zal water de cellen van de plant verlaten, en zal de plant dus uitdrogen en dood gaan. (Osmosis in Plant Cells - Transport Across Membranes - National 5 Biology Revision - BBC Bitesize, 2023)

Er zijn onderzoeken gedaan naar de invloed van zoutconcentraties op planten, echter is er niks te vinden over de invloed op kiemgroenten.

Als we kijken naar onderzoeken die al gedaan zijn bij dit onderwerp zien we de volgende concentraties:

- 0%
- 2,922%
- 5,844%
- 8,766%

Uit dit onderzoek is gebleken dat er wel degelijk een significante verandering is op de groei van plantjes, tussen de verschillende zoutconcentraties. Hoe hoger de concentratie hoe meer er een negatieve impact is op de groei van de planten. (Habibi, Armin & Abdoli, Majid. (2013). Influence of salicylic acid pre-treatment on germination, vigor and growth parameters of garden cress (*Lepidium sativum*) seedlings under water potential loss at salinity stress.)

Zo zullen we waarschijnlijk zien dat de planten met 0% zout mogelijk wel gaan groeien, maar niet veel. De 0.01% zout oplossing zal waarschijnlijk het beste groeien want die zout concentratie komt het meest overeen met de gemiddelde zoutconcentratie van planten.

Bronnen

Habibi, Armin & Abdoli, Majid. (2013). Influence of salicylic acid pre-treatment on germination, vigor and growth parameters of garden cress (*Lepidium sativum*) seedlings under water potential loss at salinity stress.

Osmosis in plant cells - Transport across membranes - National 5 Biology Revision - BBC Bitesize. (2023, 17 februari). BBC Bitesize. <https://www.bbc.co.uk/bitesize/guides/zqdhjty/revision/6>

Verwerking 7 mei

Om alvast te kijken hoe we met de data om kunnen gaan, ga ik een basic dataframe maken van 1 van de planten. In deze dataframe komt alleen maar de groei van de plant en de oplossing zout te staan. Met deze data worden verschillende data-outputs gemaakt. Het test dataframe is opgesteld door Ramon, tot en met n_blaadjes, het laatste deel door mij.

Logboek kiemgroenten

2024-05-07

Om alvast te kijken hoe we met de data om kunnen gaan, ga ik een basic dataframe maken van 1 van de planten. In deze dataframe komt alleen maar de groei van de plant en de oplossing zout te staan. Met deze data wordt een histogram gemaakt.

Dataset

```
library(conflicted)
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2    3.5.1      v tibble     3.2.1
## v lubridate  1.9.3      v tidyr      1.3.1
## v purrr      1.0.2

library(gridExtra)
library(pwr)

set.seed(0)
cress_data <- data.frame(species = rep(c("cress", "alfalfa"), each=400),
```



```
## 3 cress      3      0      6.99      1      3.82
## 4 cress      4      0      6.91      1      3.11
## 5 cress      5      0      5.62      0      3.76
## 6 cress      6      0      2.69      0      0.694
```

```
# Gemiddelde generen voor groei in cm voor elke soort en oplossing
aggregate(cm_groei ~ species + Oplossing, data=cress_data, FUN=mean)
```

```
##   species Oplossing cm_groei
## 1 alfalfa      0 3.103501
## 2  cress      0 5.304877
## 3 alfalfa      1 3.835705
## 4  cress      1 7.863672
## 5 alfalfa      2 2.018409
## 6  cress      2 2.715076
## 7 alfalfa      3 1.097948
## 8  cress      3 1.124463
```

2024-05-08

Test data structuur

Omdat deze data gesimuleerd is, is dit overbodig. Het moet daarintgen wel in onze echte analyse. Dus verwerk ik dit hier nju

```
head(cress_data)
```

```
## # A tibble: 6 x 7
##   species      N batch_n Oplossing cm_groei n_ontkieming n_blaadjes
##   <chr>   <int>   <dbl>   <dbl>   <dbl>         <int>     <dbl>
## 1 cress     1       1       0     6.89           1       2.04
## 2 cress     2       1       0     4.51           1       1.38
## 3 cress     3       1       0     6.99           1       3.82
## 4 cress     4       1       0     6.91           1       3.11
## 5 cress     5       1       0     5.62           0       3.76
## 6 cress     6       1       0     2.69           0       0.694
```

De correcte datatypen komen voor in de data set, ook is de header correct.

Het volgende wat gecheckt moet worden is of het aantal observaties overeenkomt met de dataset

```
str(cress_data)
```

```
## tibble [800 x 7] (S3: tbl_df/tbl/data.frame)
## $ species      : chr [1:800] "cress" "cress" "cress" "cress" ...
## $ N            : int [1:800] 1 2 3 4 5 6 7 8 9 10 ...
## $ batch_n      : num [1:800] 1 1 1 1 1 1 1 1 1 1 ...
## $ Oplossing    : num [1:800] 0 0 0 0 0 0 0 0 0 0 ...
## $ cm_groei     : num [1:800] 6.89 4.51 6.99 6.91 5.62 ...
## $ n_ontkieming: int [1:800] 1 1 1 1 0 0 1 1 1 1 ...
## $ n_blaadjes   : num [1:800] 2.04 1.38 3.82 3.11 3.76 ...
```

800 observaties komt overeen met de 800 zaadjes die geteld en geplant zijn.

0,1,2,3 nieuwe concentraties

2024-05-09

- 5 liter water gekookt voor 5 minuten.

- 5, 10, 15 gram zout gewogen
- deze heb ik opgelost in 3x500 ml water
- dit zorgt voor 3 verschillende concentraties: 1,2,3% zout
- Deze oplossingen zijn in een eigen spuitfles gedaan.
- 500 ml water zonder opgeloste zout ook in een eigen spuitfles gedaan.
- 16 petrischaaltjes gelabeld met soort, batch n en % oplossing
- 1 keukenpapiertje per schaalte 2x dubbel gevouwen en gespoten met de oplossing die bij het schaalte hoort tot het vochtig is aan de bovenkant.
- keukenpapiertje in schaalte gedaan en 50 zaadjes er over heen gestrooid.
- zaadjes gespoten met correcte oplossing water na het het strooien.

2024-05-10

- 8:40: papier was droog, dus heb elk bakje 2x bemist ### t-test test Een goede manier om voor elke oplossing een t test te doen tussen de 2 soorten

```
cress_data %>%
  dplyr::filter(Oplossing == 0) %>%
  t.test(cm_groei ~ species, data=.)
```

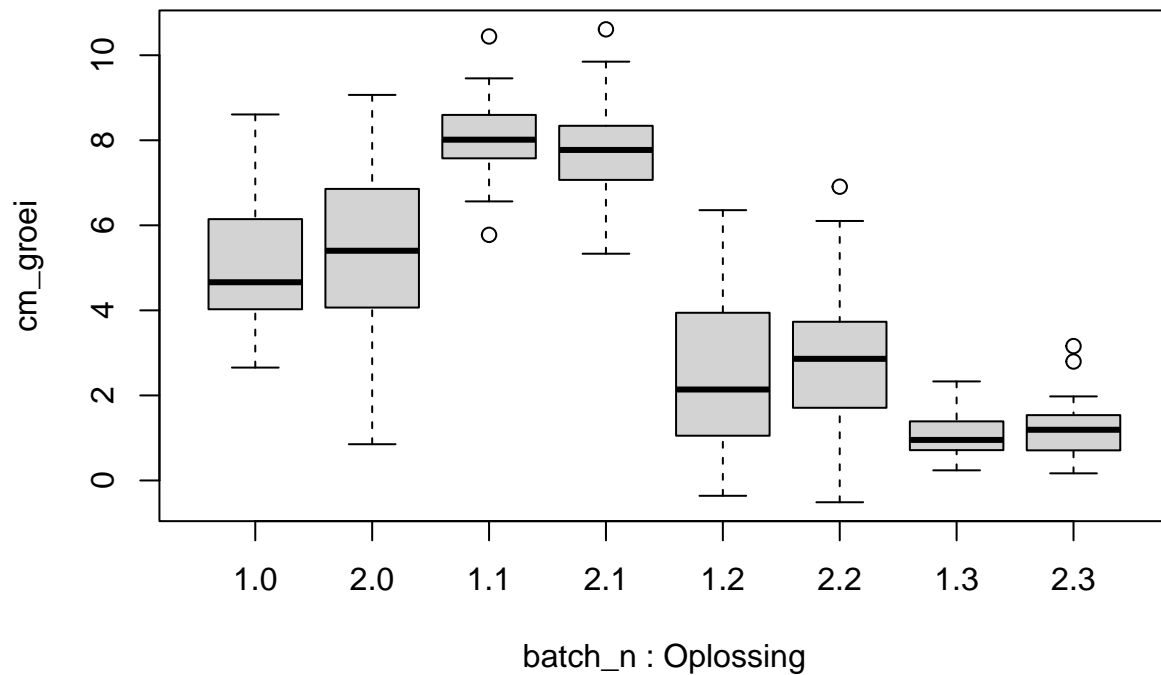
```
##
## Welch Two Sample t-test
##
## data: cm_groei by species
## t = -11.251, df = 179.71, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group alfalfa and group cress is not equal
## 95 percent confidence interval:
## -2.587451 -1.815301
## sample estimates:
## mean in group alfalfa mean in group cress
## 3.103501 5.304877
```

Deze manier is niet heel handig, en onjuist. De juiste manier om dit uit te voeren voor onze data set is een 2-way anova test. Deze krijgen wij pas in een latere les, en zal dus ook later gebruikt worden.

standaard boxplot showcase

Om de spreiding weer te geven tussen de 2 batches op elke oplossing van de cress kan je een boxplot gebruiken, zie hieronder

```
test <- cress_data %>%
  dplyr::filter(species == "cress")
boxplot(cm_groei ~ batch_n + Oplossing, data=test)
```



2024-05-11

- Omdat het papier uitdroogt spuit ik elke ochtend en avond om 9 uur 2x in elk bakje. Ook laat ik het dekseltje met een klein spleetje open op het bakje, zodat het vocht in het bakje blijft

2024-05-12

Vanwege een fout tijdens het 1e experiment, wat heeft geleid tot extreme uitdroging hebben we besloten overnieuw te beginnen

- zelfde stappen als voorheen
- nu vanaf het begin het dekseltje op het bakje gelaten, zodat de bodem niet uitdroogt.

2024-05-13

- 10 uur in de ochtend 2 keer water gespoten op de zaadjes

sapply test

Er moet vast een manier zijn om een functie meerdere keren te herhalen, en die data op te slaan. Om dit te testen experimenteer ik met lapply, met het uitvoeren van meerdere t-testen

```
# Zo kan je voor elke oplossing kijken of er een significante afwijking is in verhouding met 0%
lijstje <- sapply(1:3, function(x){
  y <- cress_new %>%
    dplyr::filter(Oplossing == 0 | Oplossing == x) %>%
    dplyr::filter(species == "cress") %>%
```

```

t.test(cm_groei ~ Oplossing, data = .)})

# Dit gebeurt er boven ^
x1 <- cress_data %>%
  dplyr::filter(Oplossing == 0 | Oplossing == 1) %>%
  dplyr::filter(species == "cress") %>%
  t.test(cm_groei ~ Oplossing, data = .)

x2 <- cress_data %>%
  dplyr::filter(Oplossing == 0 | Oplossing == 2) %>%
  dplyr::filter(species == "cress") %>%
  t.test(cm_groei ~ Oplossing, data = .)

x3 <- cress_data %>%
  dplyr::filter(Oplossing == 0 | Oplossing == 3) %>%
  dplyr::filter(species == "cress") %>%
  t.test(cm_groei ~ Oplossing, data = .)

```

Zo krijg je als output de t-test uitslagen van het verschil van cm_groei tussen oplossing van 0% en de overige oplossingen. Ook deze manier is niet relevant voor onze dataset, een 2 way anova test zou beter hiervoor zijn. Dit was een test om de sapply functie te kunnen gebruiken. De 2-way anova test komt later in het logboek.

2024-05-14

- Plantjes water gegeven om 9 uur.

Proportie

Vandaag hebben we geleerd hoe we om kunnen gaan met percentages, dit is relevant voor onze ontkieming.

```

data_set_ontkieming <- cress_data %>%
  # Selecteer de juiste kolommen
  select(n_ontkieming, species, Oplossing) %>%
  mutate(species_oplossing = fct_cross(species, as.character(Oplossing), sep = "-"))
head(data_set_ontkieming)

```

```

## # A tibble: 6 x 4
##   n_ontkieming species Oplossing species_oplossing
##         <int> <chr>      <dbl> <fct>
## 1             1 cress         0 cress-0
## 2             1 cress         0 cress-0
## 3             1 cress         0 cress-0
## 4             1 cress         0 cress-0
## 5             0 cress         0 cress-0
## 6             0 cress         0 cress-0

```

Maak een tabel aantal ontkiemde zaadjes voor de 2 soorten, en oplossingen

```
ontkieming_table <- table(data_set_ontkieming$species, data_set_ontkieming$Oplossing + data_set_ontkieming$species_oplossing)
```

Nu kunnen we een prop.test doen om te kijken of er verschil is tussen de 2 soorten planten

```
addmargins(ontkieming_table)
```

```

##
##           0    1    2    3    4 Sum
## alfalfa  48   67  139  112   34 400

```

```
##    cress    41  90 129 120  20 400
##    Sum      89 157 268 232  54 800
```

Om het verschil te kunnen zien tussen de 2 soorten en n ontkiemeningen kan je een 2-way chisq-test uitvoeren:

```
# 2-way chisq uitvoeren om verschil te kunnen testen
(res <- chisq.test(x = ontkieming_table))
```

```
##
## Pearson's Chi-squared test
##
## data:  ontkieming_table
## X-squared = 8.1986, df = 4, p-value = 0.08457
cat(sprintf("De p-waarde is groter dan 0.05: %.2f\n", res$p.value))
```

```
## De p-waarde is groter dan 0.05: 0.08
```

De p-waarde is groter dan 0.05, dit houdt in dat er geen significant verschil is tussen de 2 planten soorten. Om te weten of het een relevant verschil is, moet je de effectsterkte berekenen. Voor proportie gebruik ik Cohen's W

```
chi2 <- res$statistic
N <- sum(ontkieming_table)
W <- sqrt(chi2 / N)
df <- res$parameter
cat(sprintf("Cohens W: %.3f", W))
```

```
## Cohens W: 0.101
```

Cohens W ligt bij 0.10, er is dus een zwak verband tussen de invloed op aantal ontkiemde zaadjes tussen soort plant en zoutoplossing.

power

Nu we een conclusie hebben moeten we berekenen of die correct is of niet. Dit kan door de power te berekenen:

```
alpa <- 0.05
pwr.chisq.test(w=W, N=N, df=df, sig.level = alpa)
```

```
##
##      Chi squared power calculation
##
##              w = 0.1012337
##              N = 800
##              df = 4
##      sig.level = 0.05
##      power    = 0.6168838
##
## NOTE: N is the number of observations
```

Hier is de power 1 Daarmee kunnen we zeggen dat onze conclusie, er is een significant verschil gevonden van het aantal ontkiemde zaadjes tussen de 2 planten over de verschillende oplossingen, een 100% kans heeft om correct te zijn.

Hoeveel metingen hebben we nodig om 0.8 power te krijgen, met een W van 0.1?

```
pwr.chisq.test(w = 0.1, df = df, sig.level = alpa, power = 0.80)
```

```
##
##      Chi squared power calculation
```



```
##
##           w = 0.1
##           N = 1193.529
##           df = 4
##       sig.level = 0.05
##           power = 0.8
##
## NOTE: N is the number of observations
```

Dan zouden we dus 1435 planten moeten meten, zo'n 600 per planten soort.

2024-05-15

- 7:30 planten water gegeven

ggplot2 plots

We hebben voor cm_groei een boxplot gemaakt via de basic boxplot functie van R. Graag zou ik alle plots willen maken via de ggplot library.

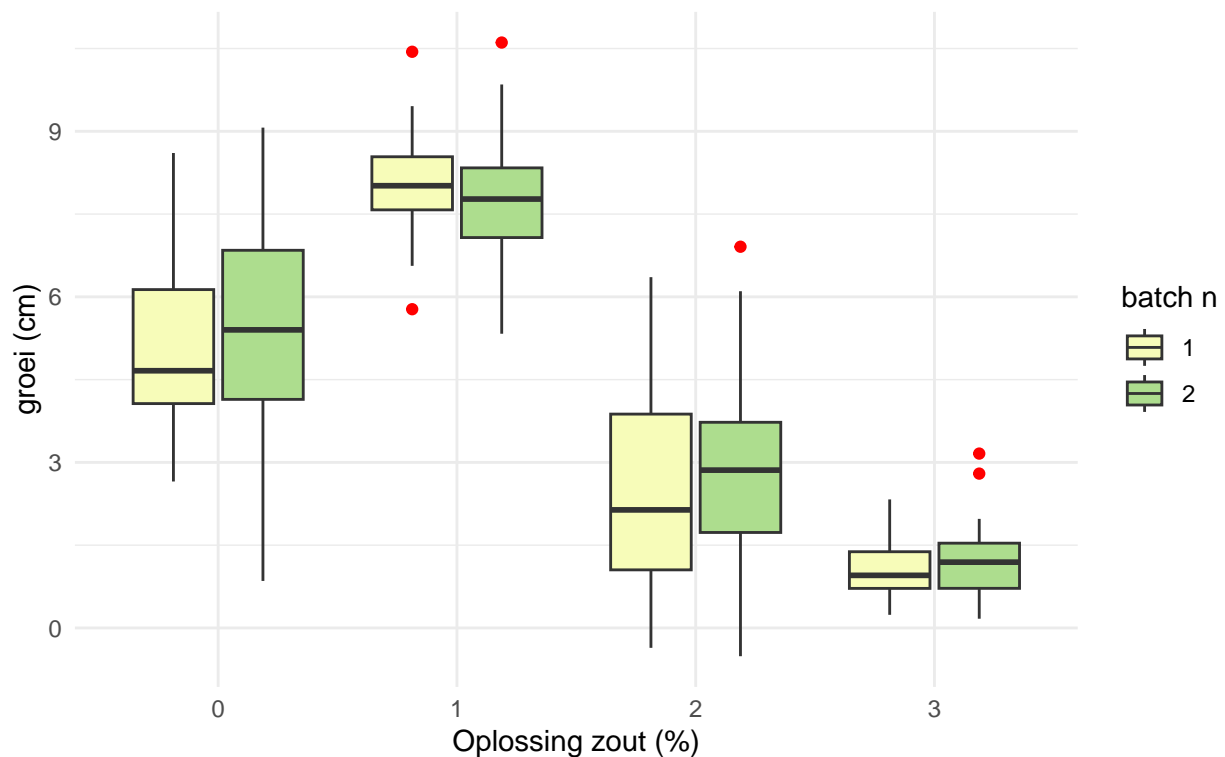
```
cress_data%>%
  # Filter data op 1 soort
  dplyr::filter(species == "cress") %>%
  ggplot(mapping = aes(x = as.character(Oplossing), y = cm_groei, fill = as.character(batch_n), ))+

  # Labels
  labs(y = "groei (cm)",
       x = "Oplossing zout (%)",
       title = "Spread of cress growth",
       subtitle = "For 0-3% salt",
       fill="batch n")+

  # Theming
  scale_fill_brewer(palette = "YlGn") +
  geom_boxplot(outlier.color = "red",) +
  theme_minimal()
```

Spread of cress growth

For 0–3% salt



Zo is de boxplot wat mooier en makkelijker af te lezen. Hetzelfde kan gedaan worden voor de andere planten soort

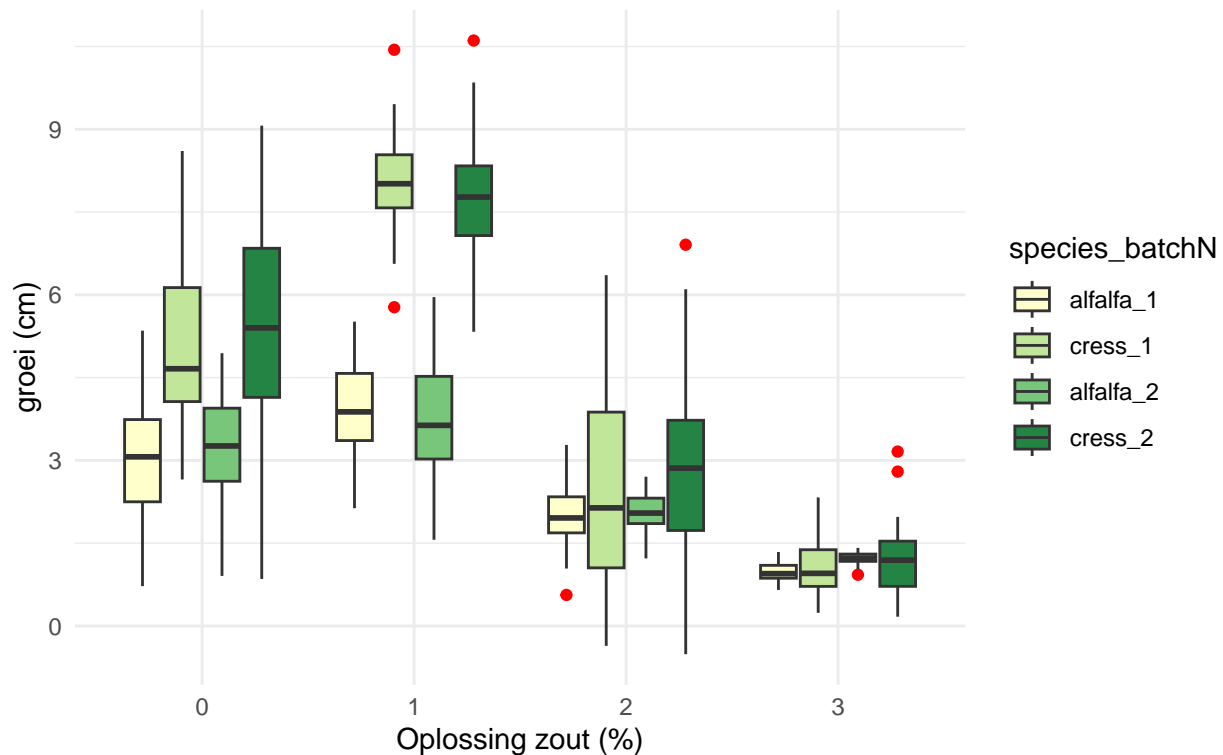
```

cress_data %>%
  # Maak nieuwe kolom die soort en batch combineert.
  mutate(species_batch = fct_cross(species, as.character(batch_n), sep = "_")) %>%
  ggplot(mapping = aes(x = as.character(Oplossing), y = cm_groei, fill = species_batch,)) +

  # Levels
  labs(y = "groei (cm)",
       x = "Oplossing zout (%)",
       title = "Spread of plant growth",
       subtitle = "For 0-3% salt",
       fill="species_batchN") +

  # Theming
  scale_fill_brewer(palette = "YlGn") +
  geom_boxplot(outlier.color = "red") +
  theme_minimal()
  
```

Spread of plant growth For 0–3% salt



Dit is best onleesbaar, vooral oplossing 3. Een betere manier om dit te doen is dit te plotten in losse charts op 1 pagina. Dit kan met een library: gridExtra

```
# Maak functie voor het plotten.
plot_data_box <- function (data, variable_oplossing){
  data %>%
    # Filter op basis van gewenste oplossing
    dplyr::filter(Oplossing == variable_oplossing) %>%
    ggplot(mapping = aes(x = species, y = cm_groei, fill=as.character(batch_n))) +

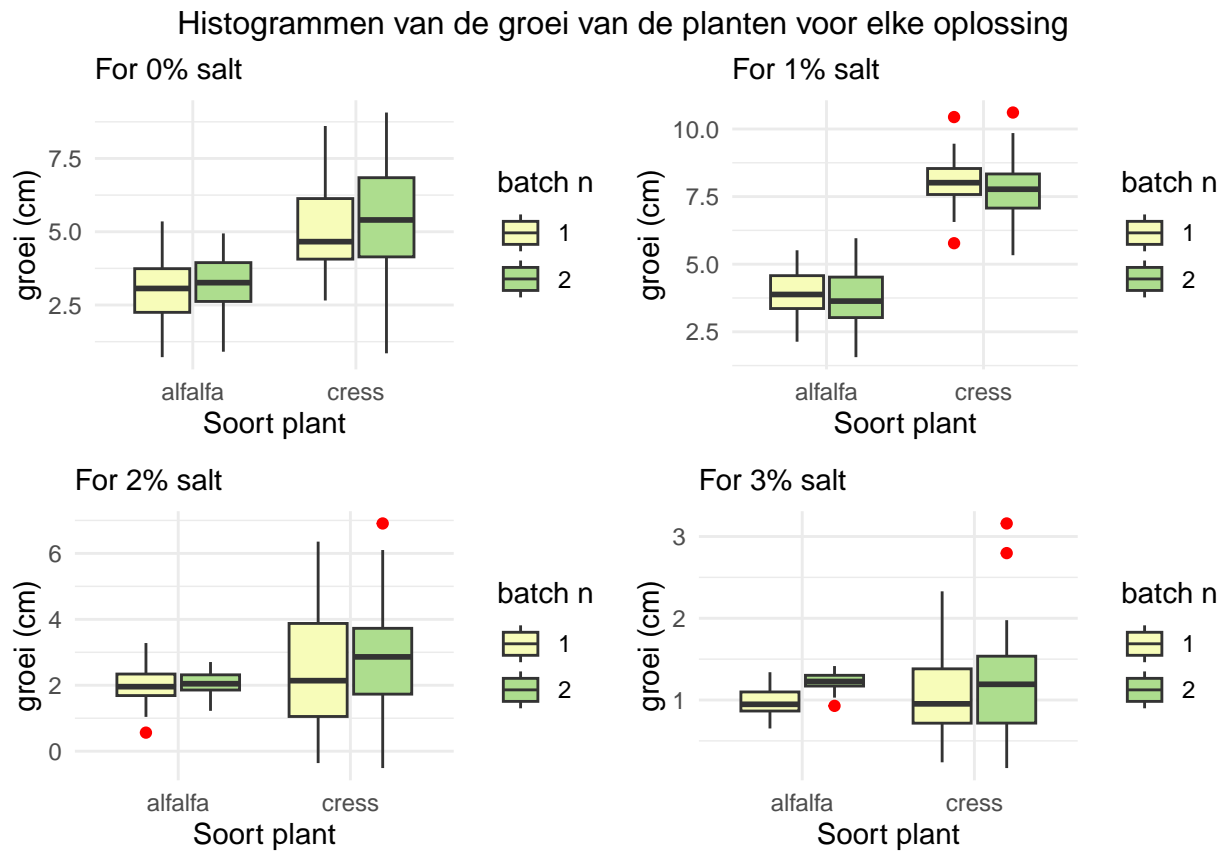
    # Labels
    labs(y = "groei (cm)",
         x = "Soort plant",
         subtitle = sprintf("For %i%% salt", variable_oplossing),
         fill="batch n") +

    # Theming
    scale_fill_brewer(palette = "YlGn") +
    geom_boxplot(outlier.color = "red")+
    theme_minimal()
}
```

Functie is gemaakt om makkelijk te plotten

```
# Gebruik gemaakt functie voor gridExtra showcase
p1 <- plot_data_box(cress_data, 0)
p2 <- plot_data_box(cress_data, 1)
p3 <- plot_data_box(cress_data, 2)
```

```
p4 <- plot_data_box(cress_data, 3)
grid.arrange(p1, p2, p3, p4, top = "Histogrammen van de groei van de planten voor elke oplossing")
```



Deze page heeft 4 boxplotjes. Voor elke oplossing geeft het de spreiding van groei in cm weer, voor de soort en batch. De rode puntjes zijn de uitschieters.

Er kan op deze manier ook een histogram gemaakt worden die het aantal ontkiemde zaadjes bevat

```
cress_data %>%
  # Maak nieuwe kolom die soort en batch combineert.
  mutate(species_batch = fct_cross(species, as.character(batch_n), sep="_")) %>%

  # Filter op een ontkiemings waarde van 1
  dplyr::filter(n_ontkieming == 1) %>%
  ggplot(mapping = aes(x = Oplossing, fill = species_batch),) +

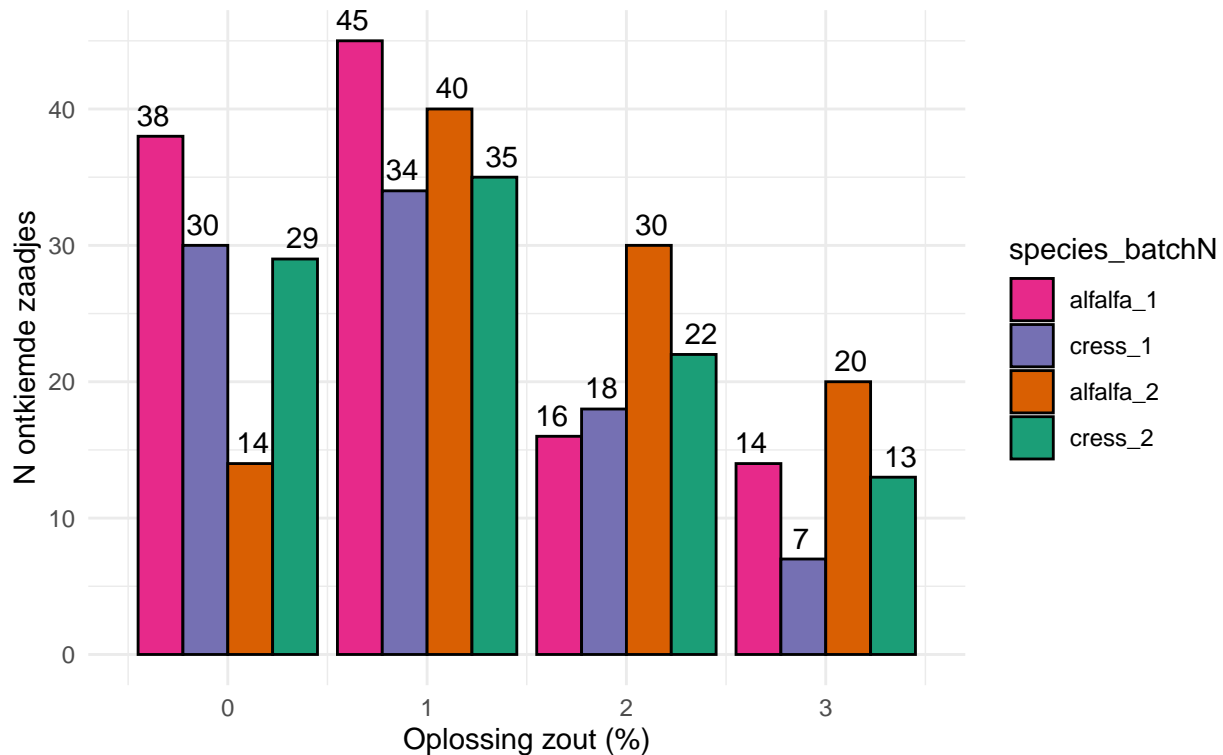
  # Labels
  labs(y = "N ontkiemde zaadjes",
       x = "Oplossing zout (%)",
       title = "Histogram van N ontkiemde zaadjes",
       subtitle = "For 0-3% salt",
       fill="species_batchN") +

  # Theming
  scale_fill_brewer(palette = "Dark2", direction = -1) +
  geom_bar(position = "dodge", color = "black")+
  stat_bin(binwidth = 1, geom = "text", aes(label = after_stat(count)), vjust=-0.5, position = "dodge")
```

```
theme_minimal()
```

Histogram van N ontkiemde zaadjes

For 0–3% salt



In deze grafiek zie je het aantal ontkiemde zaadjes per oplossing, voor alle batches en plant soorten.

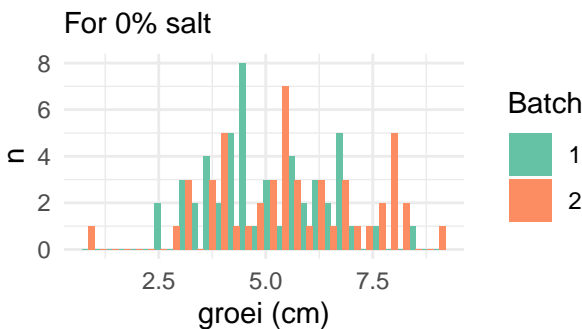
```
plot_data_hist <- function (data, variable_oplossing, v_species){  
  data %>%  
    # Filter voor gewenste oplossing en soort  
    dplyr::filter(Oplossing == variable_oplossing & species == v_species) %>%  
    ggplot(mapping = aes(x = cm_groei, fill=as.character(batch_n))) +  
  
    # Labels  
    labs(y = "n",  
         x = "groei (cm)",  
         title = sprintf("Hist of growth of %s\n", v_species),  
         subtitle = sprintf("For %i%% salt", variable_oplossing),  
         fill="Batch")+  
  
    # Theming  
    scale_fill_brewer(palette = "Set2")+  
    geom_histogram(position = "dodge",)+  
    theme_minimal()}
```

```
p1 <- plot_data_hist(cress_data, 0, "cress")  
p2 <- plot_data_hist(cress_data, 1, "cress")  
p3 <- plot_data_hist(cress_data, 2, "cress")  
p4 <- plot_data_hist(cress_data, 3, "cress")
```

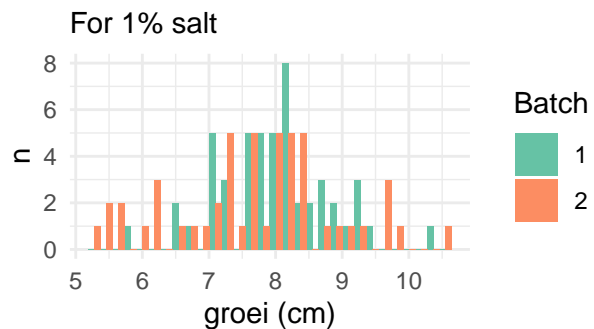
```
grid.arrange(p1, p2, p3 ,p4)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

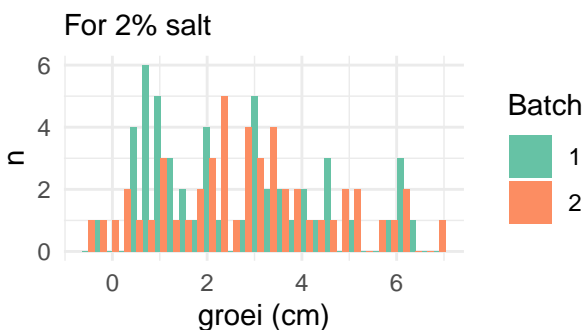
Hist of growth of cress



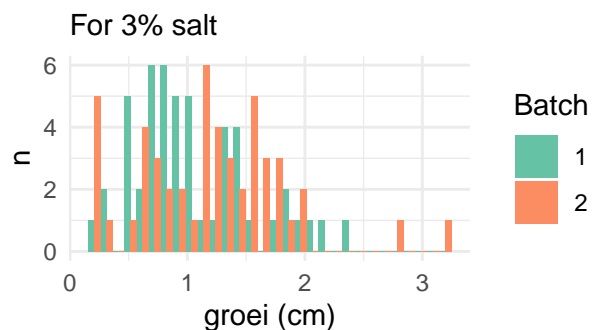
Hist of growth of cress



Hist of growth of cress



Hist of growth of cress



Hierboven een histogram van de tuinkers, deze histogram geeft de groei in cm weer voor elke zoutoplossing. Ook kan je het verschil zien tussen de 2 batches. Deze is alleen niet heel erg leesbaar vind ik. Doordat de 2 batches door elkaar lopen kan het snel verwarrend worden. Het beste is om de batches ook los van elkaar te plotten.

```
# Maak dataframe die av berekend voor cm_groei en n_blaadjes voor soort, batch en oplossing
averages_data_frame <- aggregate(cbind(cm_groei, n_blaadjes) ~ species+batch_n+Oplossing, data = cress_data, FUN = mean)

# Geef kolommen een betere naam
colnames(averages_data_frame) <- c("species", "batch_n", "Oplossing", "av_cm_groei", "av_n_blaadjes")

# Maak df die sd berekend voor cm_groei en n_blaadjes voor elke soort, batch en oplossing
temp <- aggregate(cbind(cm_groei, n_blaadjes) ~ species+batch_n+Oplossing, data = cress_data, FUN = sd)

# Voeg df samen
averages_data_frame$sd_cm_groei <- temp$cm_groei
averages_data_frame$sd_n_blaadjes <- temp$n_blaadjes
head(averages_data_frame)
```

```
## species batch_n Oplossing av_cm_groei av_n_blaadjes sd_cm_groei sd_n_blaadjes
## 1 alfalfa 1 0 2.979804 3.087327 1.2132633 1.066368
```

## 2	cress	1	0	5.035896	3.010634	1.3744421	0.945506
## 3	alfalfa	2	0	3.227199	3.081805	1.0631705	1.066072
## 4	cress	2	0	5.573858	2.886641	1.7504073	1.004672
## 5	alfalfa	1	1	3.938455	4.000246	0.7917711	1.069972
## 6	cress	1	1	8.021406	4.016331	0.8569964	1.052283

Zo krijg je een dataframe met de gemiddelden en standaard deviaties van cm_groei, en n_blaadjes.

2024-05-16

Facet-wrap gebruiken om makkelijker meerdere plotjes te plotten in 1 chart.

```
cress_data %>%

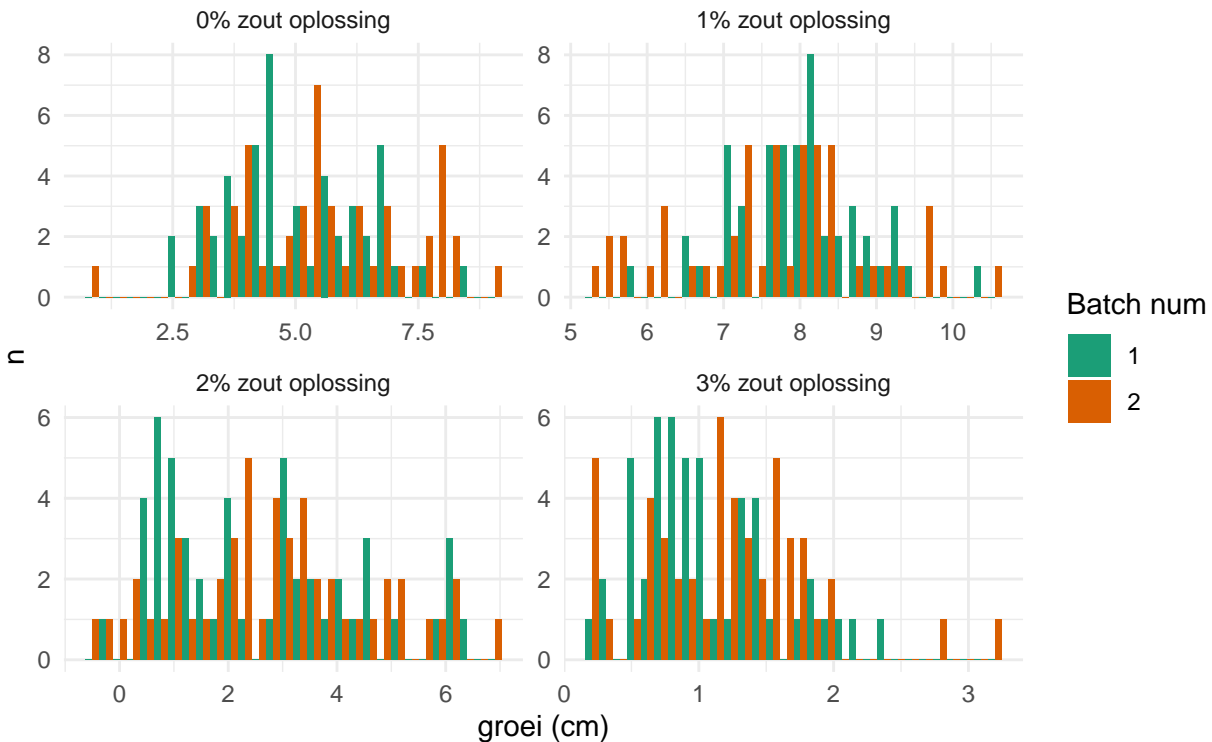
  # Filter voor gewenste soort
  dplyr::filter(species == "cress") %>%
  ggplot(mapping = aes(x = cm_groei, fill=as.character(batch_n))) +

  # Labels
  labs(y = "n",
       x = "groei (cm)",
       title = sprintf("Plant groei tuinkers"),
       subtitle = sprintf("voor zout oplossing 0-3%"),
       fill="Batch num")+

  # Theming
  scale_fill_brewer(palette = "Dark2")+
  geom_histogram(position = "dodge",) +
  facet_wrap(~ Oplossing,
             labeller = labeller(Oplossing = c("0" = "0% zout oplossing",
                                                "1" = "1% zout oplossing",
                                                "2" = "2% zout oplossing",
                                                "3" = "3% zout oplossing")),
             scales = "free")+
  theme_minimal()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Plant groei tuinkers voor zout oplossing 0–3%



Hetzelfde kan gedaan worden met de eerder gemaakte boxplotjes. Ook is deze histogram niet heel erg leesbaar, en zouden de batches ook los van elkaar geplot moeten worden. Of de batches kunnen samen genomen worden

```

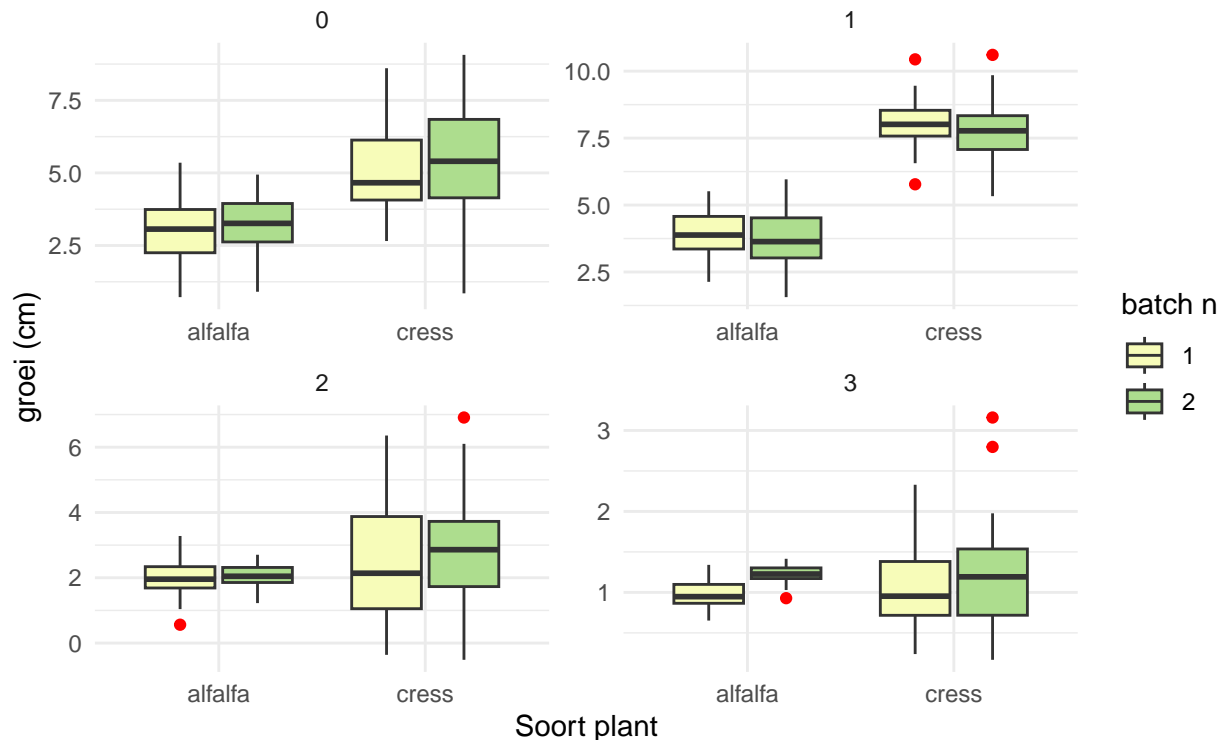
cress_data %>%
  ggplot(mapping = aes(x = species, y = cm_groei, fill=as.character(batch_n))) +

  # Labels
  labs(y = "groei (cm)",
       x = "Soort plant",
       title = "Spread of plant growth",
       subtitle = "For 0-3% salinity",
       fill="batch n") +

  # Theming
  scale_fill_brewer(palette = "YlGn") +
  geom_boxplot(outlier.colour = "red") +
  theme_minimal()+
  facet_wrap(~ Oplossing, scales = "free")

```


Spread of plant growth For 0–3% salinity



zo zijn 4 plots, zonder nog een externe library in 1 chart gedaan. Zo is het mooi overzichtelijk, en kan je snel het verschil in spreiding zien tussen batches, soorten en zoutoplossingen.

- 22:00, 0.33, 0.62, 0.90 gram zout opgelost in 3x1 liter water.
- 50 N zaadjes in bakjes gedaan voor de soorten en oplossingen in duplo.
- 3 keer de bijbehorende oplossing in het bakje gesproeid.
- bakje met deksel op vensterbank gezet.

2024-05-17

- 9:00 plantjes water gegeven

Na de les van gisteren hebben we geleerd om de 2 way anova test toe te passen. Deze test kan gebruikt worden om het verschil weer te geven tussen 2 factoren, dat geldt voor ons: 2 soorten en 4 oplossingen

Voer een 2-way anova test uit.

```
model <- aov(cm_groei ~ species+Oplossing + species*Oplossing, data = cress_data)
(res <- summary(model))
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## species         1  604.2   604.2   224.38 <2e-16 ***
## Oplossing        1 1628.7  1628.7   604.80 <2e-16 ***
## species:Oplossing 1  242.8   242.8    90.18 <2e-16 ***
## Residuals      796 2143.6     2.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Uit deze test blijkt dat: - Er een significante interactie is tussen de planten soort en zout concentratie, de p-waarde < 0.05. Omdat deze interactie significant is, houd in dat zowel soort als zout concentratie beide een

significant effect hebben op de groei van de plant.

Effect sterkte.

```
SS.a <- res[[1]]$Sum[1]
SS.b <- res[[1]]$Sum[2]
SS.ab <- sum(res[[1]]$Sum[3])
SS.tot <- sum(res[[1]]$Sum)
```

```
eta2_a <- SS.a / SS.tot
eta2_b <- SS.b / SS.tot
eta2_ab <- SS.ab / SS.tot
```

```
cat("Effect sterkten:\n")
```

```
## Effect sterkten:
```

```
cat(sprintf("Effect sterkte: eta2_a (soort): %.3f\n", eta2_a))
```

```
## Effect sterkte: eta2_a (soort): 0.131
```

```
cat(sprintf("Effect sterkte: eta2_b (zout): %.3f\n", eta2_b))
```

```
## Effect sterkte: eta2_b (zout): 0.353
```

```
cat(sprintf("Effect sterkte: eta2_ab (soort*zout): %.3f\n", eta2_ab))
```

```
## Effect sterkte: eta2_ab (soort*zout): 0.053
```

We zien de volgende dingen: - De interactie van soort*zout heeft een matig effect sterkte ($\eta_{2_ab} \sim 0.06$) op de groei van planten. Dit betekent dat er wel een interactie is tussen oplossing en soort. soort en oplossing hebben een sterk effect op de groei ($\eta_{2_a} \sim 0.14$, $\eta_{2_b} > 0.14$).

- 20:00 Plantjes water gegeven.

2024-05-18

- 9:00 plantjes water gegeven. Aantal ontkiemeningen gezien, bij alle % Gedurende de dag elke 2 uur gekeken en water gegeven op basis van de droogte.

2024-05-19

Deksel moesten van een aantal bakjes om plant groei niet te belemmeren. Elke 2 uur gecontroleerd op droogte, en water gegeven op basis van die droogte.

2024-05-20

Elke 2 uur gecontroleerd op droogte, en water gegeven op basis van die droogte.

2024-05-21

- 9:00 en 11:00 water gegeven.

```
cress_data %>%
  dplyr::filter(species == "cress") %>%
  ggplot(mapping = aes(x = cm_groei)) +

  labs(y = "n",
        x = "groei (cm)",
```

```

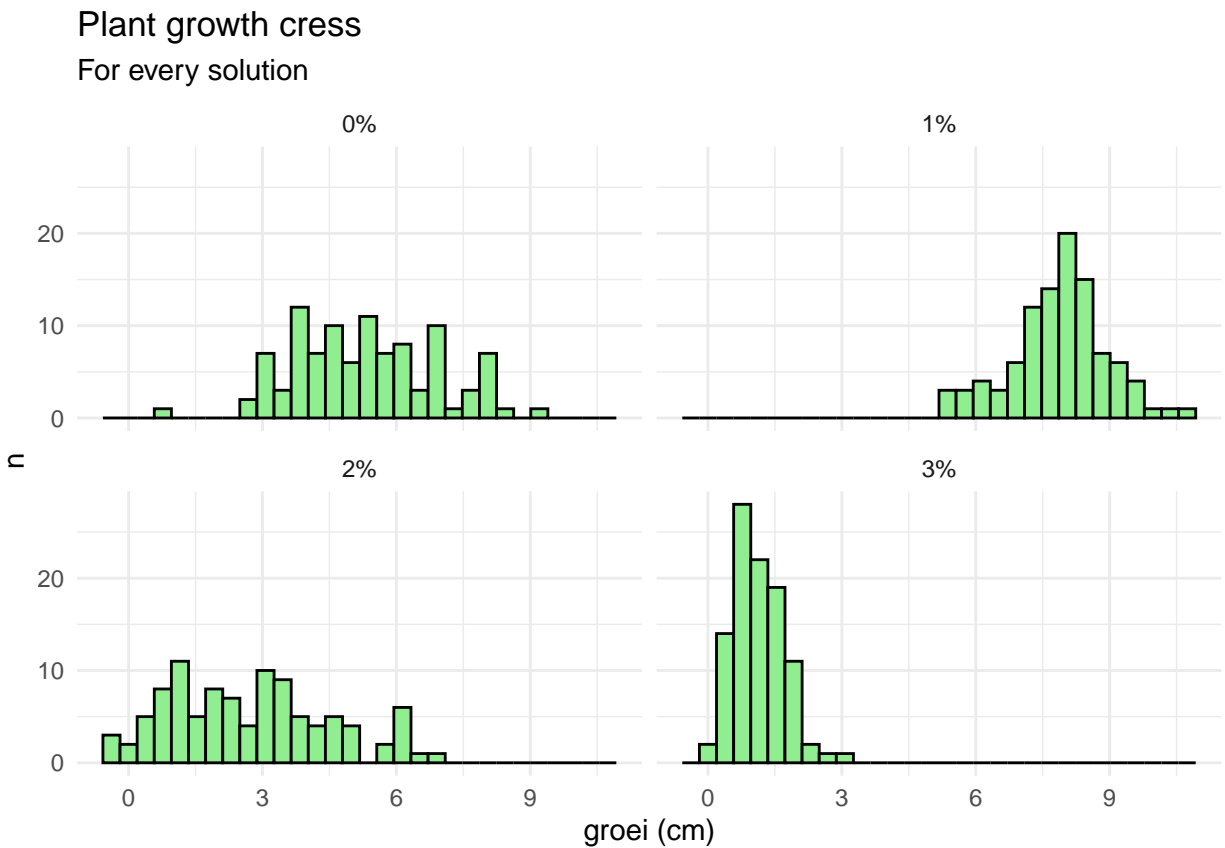
title = "Plant growth cress",
subtitle = "For every solution",) +

geom_histogram(color = "black", fill = "lightgreen")+
facet_wrap(~ Oplossing,
            labeller = labeller(Oplossing = c("0" = "0%",
                                              "1" = "1%",
                                              "2" = "2%",
                                              "3" = "3%"),
                                scales = "free"))+

theme_minimal()

```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Als je de batches samen neemt is het duidelijker af te lezen. Je ziet in deze grafiek dat de planten het meeste groeien bij 1%, en het slechtste bij 3%.