

# WK8

2025-08-21

```
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    3.5.2      v tibble    3.2.1
## v lubridate  1.9.4      v tidyr     1.3.1
## v purrr      1.0.4
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readxl)
library(lubridate)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##   chisq.test, fisher.test
```

```
library(purrr)
library(readr)
library(ggthemes)
library(ggeffects)
library(lme4)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##   expand, pack, unpack
```

```
library(dplyr)
library(ggplot2)
#install.packages("emmeans")
library(emmeans)
```

```
## Welcome to emmeans.
## Caution: You lose important information if you filter this package's results.
## See '? untidy'
```

```
#install.packages("broom.mixed")
library(broom.mixed)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:lme4':
##
##     lmList
##
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(mgcv)
```

```
## This is mgcv 1.9-3. For overview type 'help("mgcv-package")'.
```

```
#setwd("C:/Users/Tobyz/Desktop/S.tort-light-growth/Data")
```

```
import plant data
```

```
plant <- read.csv("Data/WL2-2023_Size_Combined.csv") %>%
  clean_names() %>%
  mutate(survey_date = as.Date(survey_date, format = "%m/%d/%Y"))
summary(plant)
```

```
##   survey_date      block      genotype      pop_mf
##   Min.   :2023-07-03 Length:17336 Length:17336 Length:17336
##   1st Qu.:2023-08-02 Class :character Class :character Class :character
##   Median :2023-08-30 Mode  :character Mode  :character Mode  :character
##   Mean   :2023-08-28
##   3rd Qu.:2023-09-20
##   Max.   :2023-10-20
##
##   parent_pop      mf      rep      height_cm
##   Length:17336 Min.   : 1.000 Min.   : 1.000 Min.   : 0.100
##   Class :character 1st Qu.: 2.000 1st Qu.: 4.000 1st Qu.: 1.700
##   Mode  :character Median : 5.000 Median : 8.000 Median : 3.100
##   Mean   : 4.584 Mean   : 7.932 Mean   : 4.491
##   3rd Qu.: 6.000 3rd Qu.:11.000 3rd Qu.: 5.700
```

```
##           Max.      :14.000   Max.      :27.000   Max.      :39.400
##                                     NA's      :8762
##   long_leaf_cm   survey_notes
##   Min.      :0.100   Length:17336
##   1st Qu.:1.600   Class :character
##   Median :2.500   Mode  :character
##   Mean      :2.599
##   3rd Qu.:3.500
##   Max.      :9.000
##   NA's      :9350
```

*consolidate light measurement to a weekly measurement*

```
#import light data
light_raw <- read_csv("Data/IntBioHalfHourTable_clean.txt")
```

```
## Rows: 4063 Columns: 139
## -- Column specification -----
## Delimiter: ","
## dbf  (138): RECORD, BattV_Max, PTemp_C_Max, SlrW_Avg, SlrW_Max, SlrW_Min, Sl...
## dtm   (1): TIMESTAMP
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
#weekly measurement
weekly_light <- light_raw %>%
  mutate(
    timestamp = ymd_hms(TIMESTAMP),
    SlrW_Avg = as.numeric(SlrW_Avg), # turn into number format
    week = floor_date(timestamp, "week")
  ) %>%
  group_by(week) %>%
  summarise(
    weekly_avg_SlrW = mean(SlrW_Avg, na.rm = TRUE),
    .groups = "drop"
  )
```

```
## Warning: There was 1 warning in 'mutate()'.
## i In argument: 'timestamp = ymd_hms(TIMESTAMP)'.
## Caused by warning:
## ! 84 failed to parse.
```

```
# result
print(weekly_light)
```

```
## # A tibble: 14 x 2
##   week           weekly_avg_SlrW
##   <dtm>           <dbl>
## 1 2023-07-30 00:00:00      280.
## 2 2023-08-06 00:00:00      277.
## 3 2023-08-13 00:00:00      186.
```

```
## 4 2023-08-20 00:00:00      184.
## 5 2023-08-27 00:00:00      200.
## 6 2023-09-03 00:00:00      211.
## 7 2023-09-10 00:00:00      204.
## 8 2023-09-17 00:00:00      189.
## 9 2023-09-24 00:00:00      159.
## 10 2023-10-01 00:00:00     138.
## 11 2023-10-08 00:00:00     133.
## 12 2023-10-15 00:00:00     135.
## 13 2023-10-22 00:00:00     116.
## 14 NA                      -0.616
```

*Investigate or filter out plants that show negative growth*

```
#PID
plant_growth <- plant %>%
  unite("PID", genotype:rep, sep = "_", remove = FALSE) %>%
  mutate(survey_date = as.Date(survey_date))

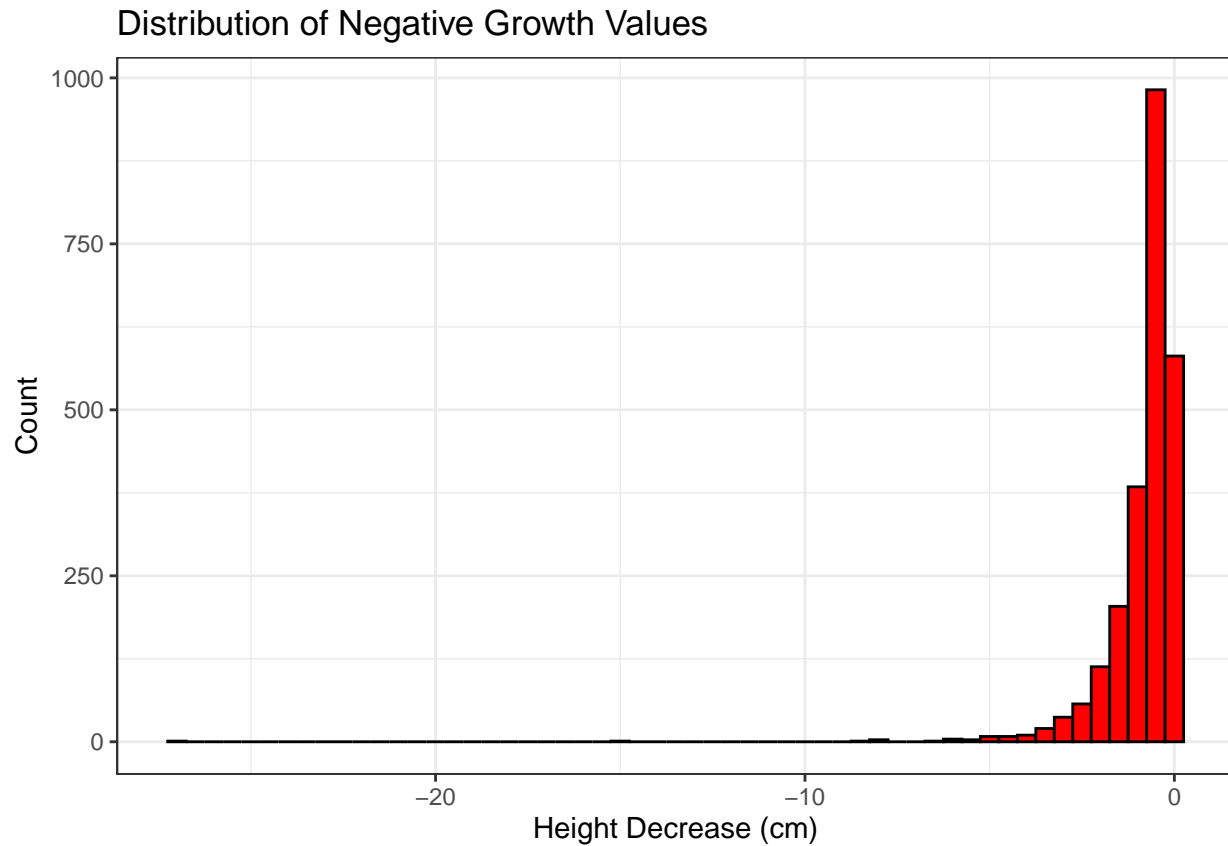
#find out plants with negative growth
plant_growth %>%
  arrange(PID, survey_date) %>% # arrange in time sequence
  group_by(PID) %>%
  mutate(growth = height_cm - lag(height_cm)) %>% # find out the diff btw nearby dates
  summarise(has_negative_growth = any(growth < 0, na.rm = TRUE)) %>%
  filter(has_negative_growth) -> neg_growth_plants
neg_growth_plants
```

```
## # A tibble: 826 x 2
##   PID                      has_negative_growth
##   <chr>                    <lgl>
## 1 BH_1_10_BH_1_BH_1_10 TRUE
## 2 BH_1_12_BH_1_BH_1_12 TRUE
## 3 BH_1_13_BH_1_BH_1_13 TRUE
## 4 BH_1_1_BH_1_BH_1_1 TRUE
## 5 BH_1_4_BH_1_BH_1_4 TRUE
## 6 BH_1_7_BH_1_BH_1_7 TRUE
## 7 BH_2_10_BH_2_BH_2_10 TRUE
## 8 BH_2_11_BH_2_BH_2_11 TRUE
## 9 BH_2_12_BH_2_BH_2_12 TRUE
## 10 BH_2_13_BH_2_BH_2_13 TRUE
## # i 816 more rows
```

```
#find out tolerance value
neg_growth_values <- plant_growth %>%
  arrange(PID, survey_date) %>%
  group_by(PID) %>%
  mutate(growth = height_cm - lag(height_cm)) %>%
  ungroup() %>%
  filter(growth < 0)

ggplot(neg_growth_values, aes(x = growth)) +
  geom_histogram(binwidth = 0.5, fill = "red", color = "black") +
```

```
labs(
  title = "Distribution of Negative Growth Values",
  x = "Height Decrease (cm)",
  y = "Count"
) +
theme_bw()
```



```
#filter out plants with negative growth < -5
plant_growth_cleaned <- plant_growth

repeat {
  plant_growth_cleaned <- plant_growth_cleaned %>%
    arrange(PID, survey_date) %>%
    group_by(PID) %>%
    mutate(growth = height_cm - lag(height_cm)) %>%
    filter(is.na(growth) | growth >= -5) %>%
    select(-growth) %>%
    ungroup()

  check <- plant_growth_cleaned %>%
    arrange(PID, survey_date) %>%
    group_by(PID) %>%
    mutate(growth = height_cm - lag(height_cm)) %>%
    filter(growth < -5)
```

```

  if (nrow(check) == 0) break
}

```

### *Measure Growth via Daily Growth Rate*

```

#define daily growth rate
plant_growth_daily <- plant_growth_cleaned %>%
  arrange(PID, survey_date) %>%
  group_by(PID) %>%
  mutate(
    prev_height = lag(height_cm),
    prev_date = lag(survey_date),
    days_elapsed = as.numeric(survey_date - prev_date),
    daily_growth = (height_cm - prev_height) / days_elapsed
  ) %>%
  ungroup()

```

### *Correlate Growth with Solar Radiation*

```

#Align plant growth data to week
plant_weekly <- plant_growth_daily %>%
  filter(!is.na(daily_growth), days_elapsed > 0) %>%
  mutate(week = floor_date(survey_date, "week"))

#Adds `weekly_avg_SlrW` to plant data
plant_with_light <- plant_weekly %>%
  left_join(weekly_light, by = "week")

#Calculate correlation
cor_result <- cor(
  plant_with_light$daily_growth,
  plant_with_light$weekly_avg_SlrW,
  use = "complete.obs"
)

print(cor_result)

```

```
## [1] 0.1299395
```

```

#Plot correlation
ggplot(plant_with_light, aes(x = weekly_avg_SlrW, y = daily_growth)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Correlation between Light and Growth", x = "Weekly Avg Light (SlrW)", y = "Daily Growth")
  theme_bw()

```

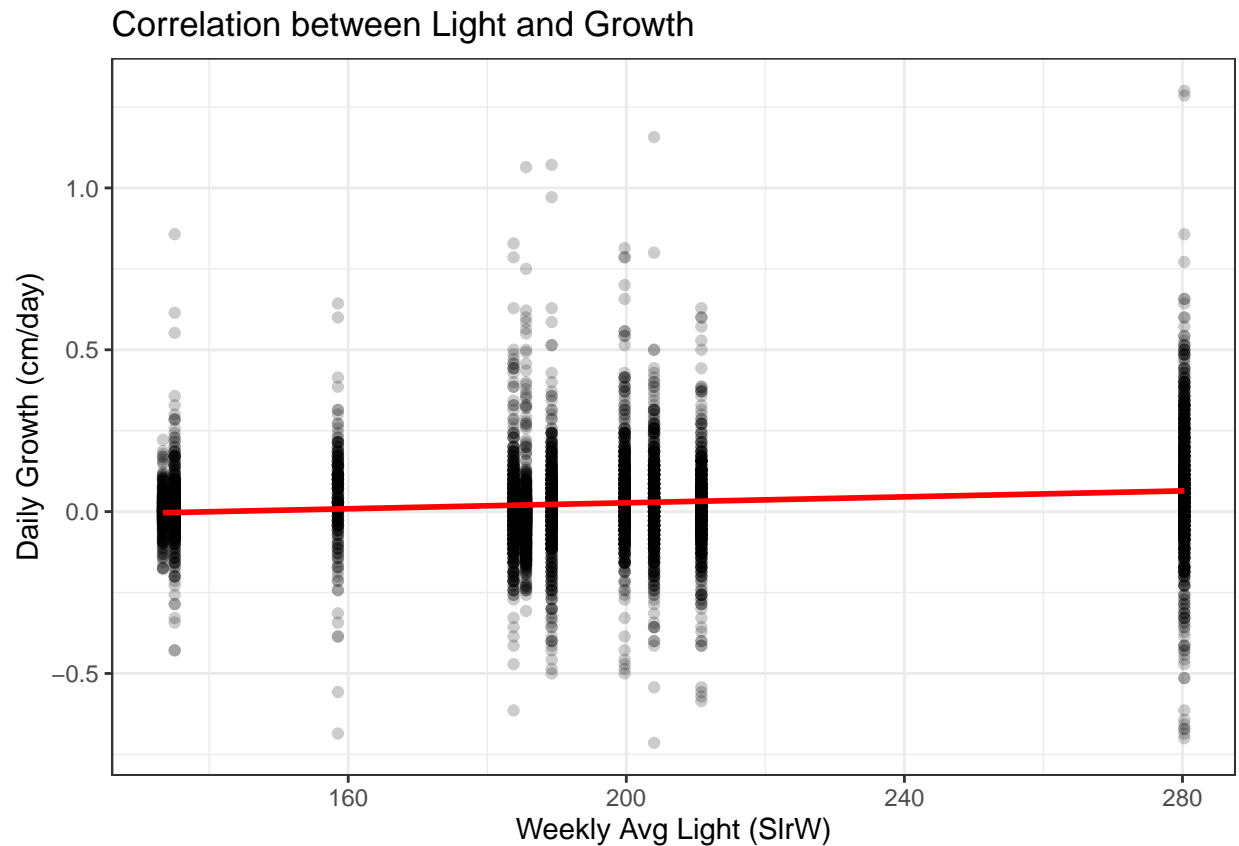
```
## 'geom_smooth()' using formula = 'y ~ x'
```

```

## Warning: Removed 1069 rows containing non-finite outside the scale range
## ('stat_smooth()').

```

```
## Warning: Removed 1069 rows containing missing values or values outside the scale range
## ('geom_point()').
```



*Process data*

```
#Standardization
plant_with_light$weekly_avg_SlrW2 <- scale(plant_with_light$weekly_avg_SlrW, center = TRUE, scale = TRUE)
mean_light <- mean(plant_with_light$weekly_avg_SlrW, na.rm = TRUE)
sd_light <- sd(plant_with_light$weekly_avg_SlrW, na.rm = TRUE)

#Change Data type
plant_with_light <- plant_with_light %>%
  mutate(
    parent_pop = factor(parent_pop),
    PID = factor(PID),
    block = factor(block)
  )
```

*use mixed-effect model to fit relationship between plant growth and light radiation with population as random effect*

```
growth_light.lmer <- lmer(
  daily_growth ~ weekly_avg_SlrW2 +
    (1 + weekly_avg_SlrW2 | parent_pop),
  data = plant_with_light, REML = TRUE
```

```

)
summary(growth_light.lmer)

## Linear mixed model fit by REML ['lmerMod']
## Formula: daily_growth ~ weekly_avg_SlrW2 + (1 + weekly_avg_SlrW2 | parent_pop)
## Data: plant_with_light
##
## REML criterion at convergence: -6464.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -6.2096 -0.4715  0.0094  0.4415  8.3380
##
## Random effects:
##   Groups      Name                Variance Std.Dev. Corr
##   parent_pop (Intercept)          0.0015837 0.0398
##              weekly_avg_SlrW2 0.0002495 0.0158  0.03
##   Residual                        0.0191023 0.1382
## Number of obs: 5870, groups: parent_pop, 22
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    0.017235  0.008784  1.962
## weekly_avg_SlrW2 0.024127  0.003973  6.073
##
## Correlation of Fixed Effects:
##              (Intr)
## wkly_vg_SW2 0.020

```

*overall effect of solar radiation on plant daily growth*

```

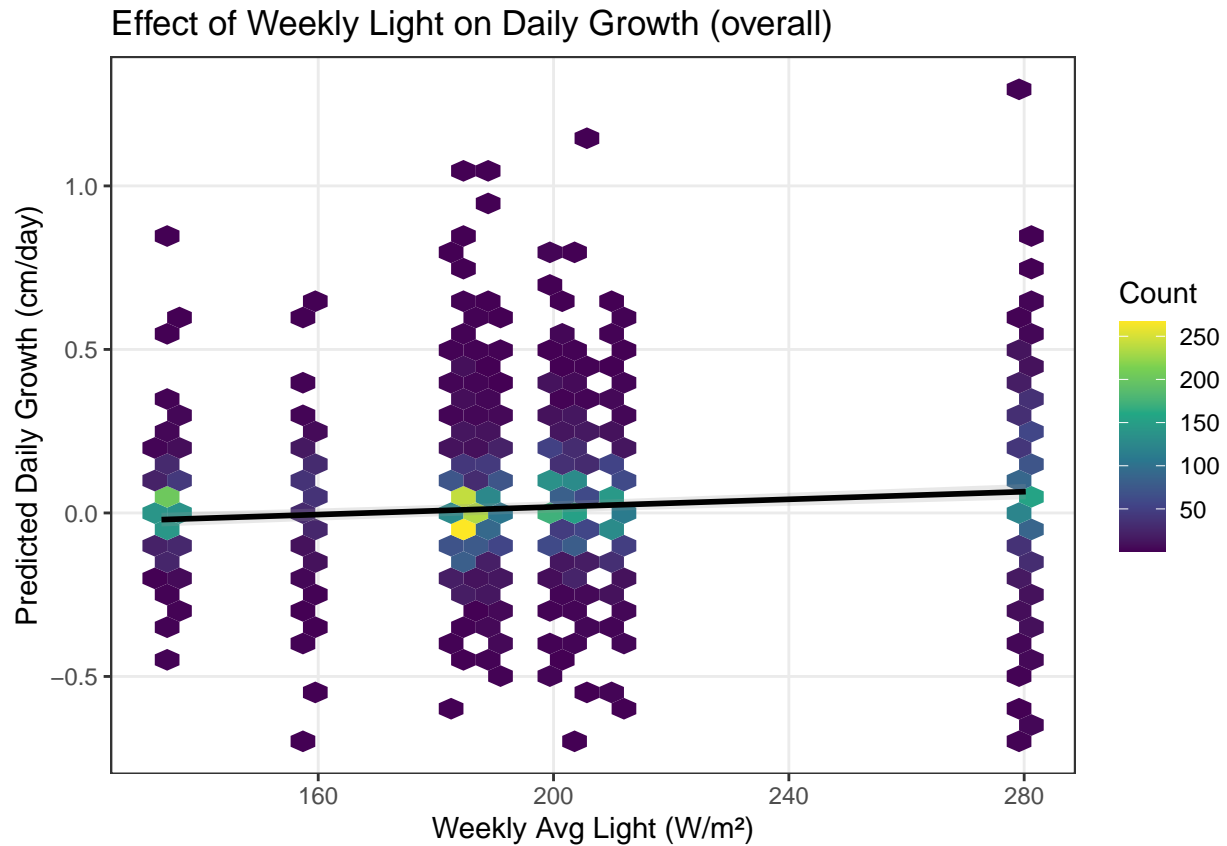
#Unscaling
pred_all <- ggpredict(growth_light.lmer, terms = "weekly_avg_SlrW2") %>%
  as.data.frame() %>%
  mutate(light_orig = x * sd_light + mean_light)

#Plot
p_overall <- ggplot() +
  geom_hex(data = plant_with_light,
    aes(x = weekly_avg_SlrW, y = daily_growth), bins = 35) +
  scale_fill_viridis_c(name = "Count") +
  geom_ribbon(data = pred_all,
    aes(x = light_orig, ymin = conf.low, ymax = conf.high),
    alpha = .22, fill = "grey60") +
  geom_line(data = pred_all,
    aes(x = light_orig, y = predicted), linewidth = 1) +
  labs(title = "Effect of Weekly Light on Daily Growth (overall)",
    x = "Weekly Avg Light (W/m²)",
    y = "Predicted Daily Growth (cm/day)") +
  theme_bw() +
  theme(panel.grid.minor = element_blank())
p_overall

```



```
## Warning: Removed 1069 rows containing non-finite outside the scale range
## ('stat_binhex()').
```



*Effect of solar radiation on plant daily growth by population*

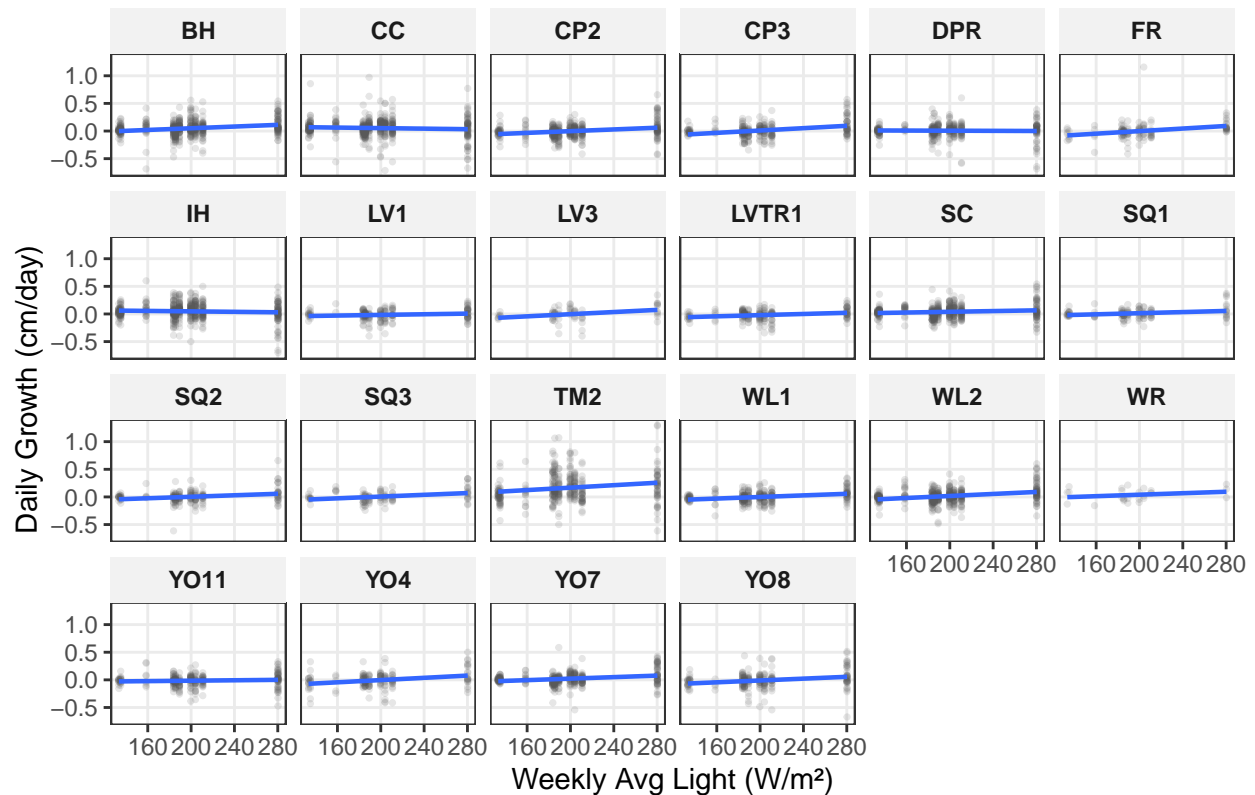
```
p_facet <- ggplot(plant_with_light,
                  aes(weekly_avg_SlrW, daily_growth)) +
  facet_wrap(~ parent_pop, ncol = 6) +
  geom_point(alpha = .15, size = .6, color = "grey35") +
  geom_smooth(method = "lm", se = FALSE, linewidth = .8) +
  labs(title = "Light-Growth relationship by population",
       x = "Weekly Avg Light (W/m²)",
       y = "Daily Growth (cm/day)") +
  theme_bw() +
  theme(strip.background = element_rect(fill = "grey95", color = NA),
        strip.text = element_text(face = "bold"),
        panel.grid.minor = element_blank())
p_facet
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: Removed 1069 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 1069 rows containing missing values or values outside the scale range
## ('geom_point()').
```

## Light–Growth relationship by population



Find slope of plant growth and solar radiation for each population

```
#slope and standard deviation of fixed effects
b_fix <- fixef(growth_light.lmer)["weekly_avg_SlrW2"]
V_fix <- vcov(growth_light.lmer)["weekly_avg_SlrW2", "weekly_avg_SlrW2"]

#slope and standard deviation of random effects
re <- ranef(growth_light.lmer, condVar = TRUE)
re_pop <- re$parent_pop
postVar <- attr(re$parent_pop, "postVar")
sl_col <- which(colnames(re_pop) == "weekly_avg_SlrW2")

#create the tibble of slopes for each population
pop_slope <- tibble(
  parent_pop = rownames(re_pop),
  rand_slope = re_pop[, "weekly_avg_SlrW2"],
  rand_var = sapply(seq_len(dim(postVar)[3]), function(i) postVar[sl_col, sl_col, i])
) %>%
  mutate(
    slope_SD = b_fix + rand_slope,
    se_SD = sqrt(V_fix + rand_var),
    lower_SD = slope_SD - 1.96*se_SD,
    upper_SD = slope_SD + 1.96*se_SD
  )

sd_light <- sd(plant_with_light$weekly_avg_SlrW, na.rm = TRUE)
```

```

pop_slope <- pop_slope %>%
  mutate(
    slope_per_Wm2 = slope_SD / sd_light,
    lower_per_Wm2 = lower_SD / sd_light,
    upper_per_Wm2 = upper_SD / sd_light
  ) %>%
  select(parent_pop, slope_per_Wm2, lower_per_Wm2, upper_per_Wm2)
pop_slope

```

```

## # A tibble: 22 x 4
##   parent_pop slope_per_Wm2 lower_per_Wm2 upper_per_Wm2
##   <chr>         <dbl>         <dbl>         <dbl>
## 1 BH           0.000742      0.000409      0.00108
## 2 CC          -0.000128     -0.000461     0.000205
## 3 CP2          0.000744      0.000376     0.00111
## 4 CP3          0.000949      0.000542     0.00136
## 5 DPR          0.0000489   -0.000316     0.000413
## 6 FR           0.000918      0.000402     0.00143
## 7 IH          -0.000105     -0.000433     0.000223
## 8 LV1          0.000373     -0.0000792    0.000825
## 9 LV3          0.000755      0.000174     0.00134
## 10 LVTR1       0.000540      0.0000734    0.00101
## # i 12 more rows

```

```

write.csv(pop_slope, "population_slopes.csv", row.names = FALSE)

```

```

#plot
ggplot(pop_slope, aes(x = reorder(parent_pop, slope_per_Wm2), y = slope_per_Wm2)) +
  geom_hline(yintercept = 0, linetype = 2) +
  geom_pointrange(aes(ymin = lower_per_Wm2, ymax = upper_per_Wm2), linewidth = .6) +
  coord_flip() +
  labs(title = "Population-specific slopes with 95% CI",
       x = "Population", y = "Slope (cm/day per W/m²)") +
  theme_bw()

```

Population-specific slopes with 95% CI

