

WK11_2022_23

2025-09-09

```
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
```

```
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)
library(emmeans)
```

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

```
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/Toby 在大学/Maloof Lab/S.tort-light-growth/Data")
```

```
import plant data
```

```
plant_2223 <- read.csv("C:/Users/Tobyz/Desktop/Toby 在大学/Maloof Lab/S.tort-light-growth/Data/UCD_2023.csv")
clean_names() %>%
mutate(survey_date = as.Date(survey_date, format = "%m/%d/%Y"))
summary(plant_2223)
```

```
##  survey_date      block      row      col
##  Min.      :2023-01-27  Length:13626  Min.      : 3.00  Length:13626
##  1st Qu.:2023-03-17   Class :character  1st Qu.:12.00   Class :character
##  Median :2023-04-20   Mode  :character  Median :22.00   Mode  :character
##  Mean    :2023-04-26                      Mean    :21.84
##  3rd Qu.:2023-06-12                      3rd Qu.:32.00
##  Max.    :2023-07-24                      Max.    :42.00
##
##      pop      mf      rep      height
##  Length:13626  Min.    : 1.000  Min.    : 1.000  Min.    : 0.100
##  Class :character  1st Qu.: 2.000  1st Qu.: 4.000  1st Qu.: 2.300
##  Mode  :character  Median : 4.000  Median : 8.000  Median : 3.850
##                      Mean   : 4.363  Mean   : 8.753  Mean   : 9.836
##                      3rd Qu.: 6.000  3rd Qu.:12.000  3rd Qu.: 7.700
##                      Max.    :12.000  Max.    :100.000  Max.    :130.000
##                      NA's    :2      NA's    :2      NA's    :10542
##  longest_leaf
##  Min.      : 0.000
##  1st Qu.: 2.100
##  Median : 3.200
##  Mean    : 3.827
##  3rd Qu.: 5.200
##  Max.    :12.700
##  NA's    :10551
```

import light data

```
light_raw_2223 <- read_csv("C:/Users/Tobyz/Desktop/Toby 在大学/Maloof Lab/S.tort-light-growth/Data/light_2023.csv")
mutate(
  date = mdy(date),
  sol_rad_ly_day = as.numeric(sol_rad_ly_day), # turn into number format
)
```

New names:

```
## Rows: 188 Columns: 33
## -- Column specification
## ----- Delimiter: "," chr
## (17): Stn Name, CIMIS Region, Date, Jul, qc...9, qc...11, qc...13, qc..... dbl
## (15): Stn Id, ETo (in), Precip (in), Sol Rad (Ly/day), Avg Vap Pres (mBa... lgl
## (1): qc...7
## 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.
## * `qc` -> `qc...7`
## * `qc` -> `qc...9`
## * `qc` -> `qc...11`
## * `qc` -> `qc...13`
## * `qc` -> `qc...15`
## * `qc` -> `qc...17`
## * `qc` -> `qc...19`
## * `qc` -> `qc...21`
## * `qc` -> `qc...23`
## * `qc` -> `qc...25`
## * `qc` -> `qc...27`
## * `qc` -> `qc...29`
## * `qc` -> `qc...31`
## * `qc` -> `qc...33`
```

Investigate or filter out plants that show negative growth #Q: How could we deal with bad observations?
 #Solution: find out tolerance value and then filter out observance data larger than the tolerance value
 #Figure A: Histogram showing the frequency of decreases in plant height between consecutive surveys.
 Most negative growth values are close to zero, likely reflecting measurement noise, while a small number of extreme decreases (≤ -5 cm) suggest errors and were removed from subsequent analyses.

```
#PID
pl_gr <- plant_2223 %>%
  unite("PID", pop:rep, sep = "_", remove = FALSE) %>%
  mutate(survey_date = as.Date(survey_date))

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

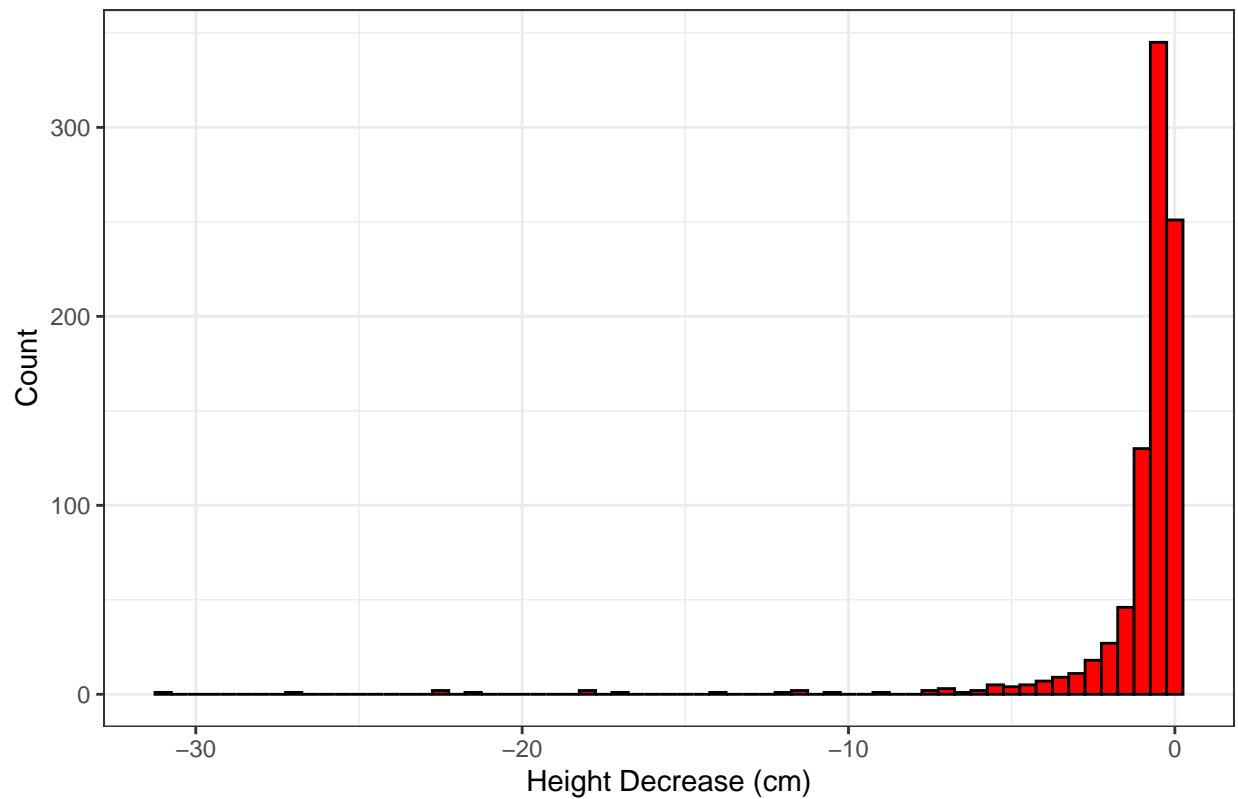
```
filter(has_negative_growth) -> neg_gr_pl
neg_gr_pl
```

```
## # A tibble: 453 x 2
##   PID      has_negative_growth
##   <chr>    <lgl>
## 1 BH_1_10 TRUE
## 2 BH_1_7  TRUE
## 3 BH_1_8  TRUE
## 4 BH_2_1  TRUE
## 5 BH_2_10 TRUE
## 6 BH_2_11 TRUE
## 7 BH_2_12 TRUE
## 8 BH_2_13 TRUE
## 9 BH_2_14 TRUE
## 10 BH_2_3 TRUE
## # i 443 more rows
```

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

ggplot(neg_gr_v, aes(x = growth)) +
  geom_histogram(binwidth = 0.5, fill = "red", color = "black") +
  labs(
    title = "Figure A: Distribution of Negative Growth Values",
    x = "Height Decrease (cm)",
    y = "Count"
  ) +
  theme_bw()
```

Figure A: Distribution of Negative Growth Values



```
#filter out plants with negative growth < -5
pl_gr_cleaned <- pl_gr

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

  check <- pl_gr_cleaned %>%
    arrange(PID, survey_date) %>%
    group_by(PID) %>%
    mutate(growth = height - lag(height)) %>%
    filter(growth < -5)

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

```
}
```

Measure Growth via Daily Growth Rate

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

Calculate Daily Mean Solar Radiation

```
pl_lt <- pl_gr_daily %>%
  rowwise() %>%
  mutate(
    mean_light_ly_day = mean(
      light_raw_2223$sol_rad_ly_day[
        !is.na(light_raw_2223$date) &
        light_raw_2223$date > prev_date &
        light_raw_2223$date <= survey_date
      ],
      na.rm = TRUE
    ),
    n_days_light = sum(
      !is.na(light_raw_2223$date) &
      light_raw_2223$date > prev_date &
      light_raw_2223$date <= survey_date,
      na.rm = TRUE
    )
  ) %>%
  ungroup()
```

Correlate Growth with Solar Radiation #Q: How does plant growth correlate with solar radiation?
#Test: Find correlation value and plot correlation #Figure B: Scatter plot showing the relationship

between daily average solar radiation (ly/day) and daily growth rate (cm/day). Each point represents an observation, and the red line indicates the fitted linear regression.

```
cor_result <- cor(
  pl_lt$daily_growth,
  pl_lt$mean_light_ly_day,
  use = "complete.obs"
)

print(cor_result)
```

```
## [1] 0.3372124
```

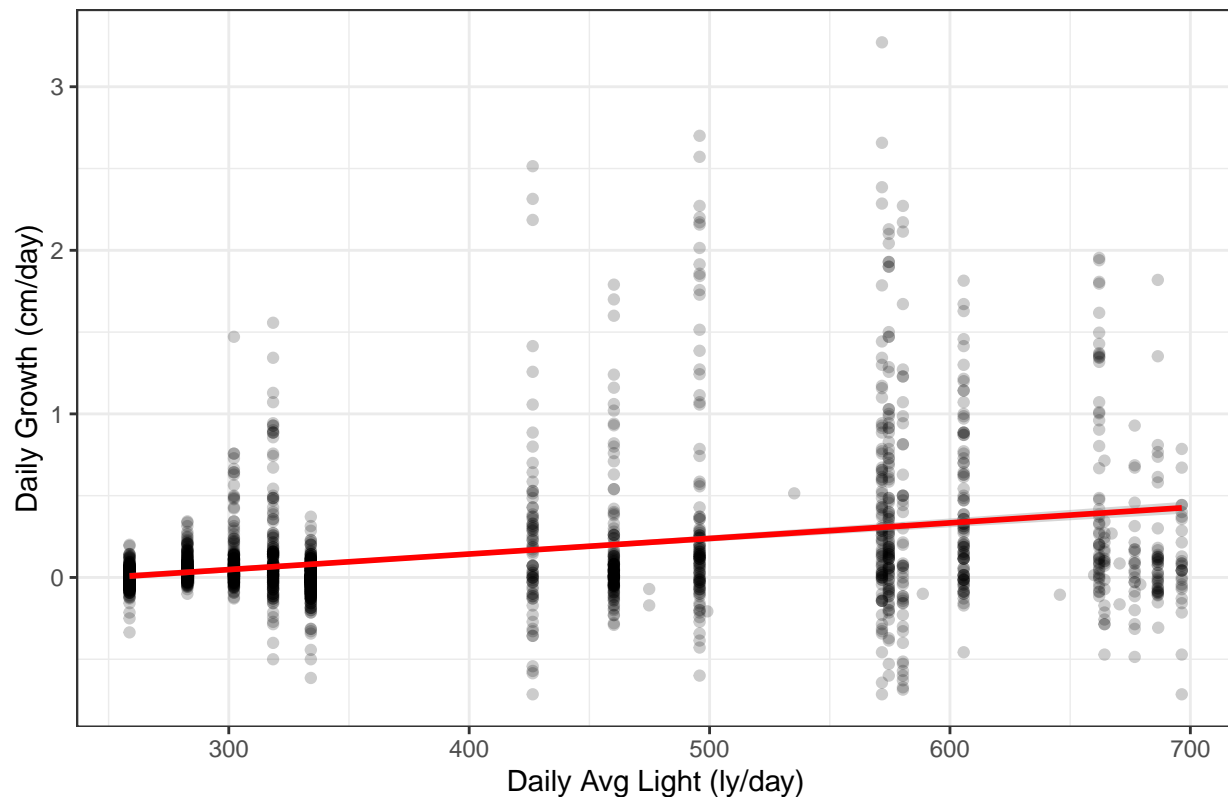
```
#Plot correlation
ggplot(pl_lt, aes(x = mean_light_ly_day, y = daily_growth)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Figure B: Correlation between Light and Growth", x = "Daily Avg Light (ly/day)", y = "Daily Growth (cm/day)") +
  theme_bw()
```

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

```
## Warning: Removed 11176 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 11176 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


Figure B: Correlation between Light and Growth



Process data

```
#Standardization
pl_lt$mean_light_ly_day2 <- scale(pl_lt$mean_light_ly_day, center = TRUE, scale = TRUE)
mean_lt <- mean(pl_lt$mean_light_ly_day, na.rm = TRUE)
sd_lt <- sd(pl_lt$mean_light_ly_day, na.rm = TRUE)

#Change Data type
pl_lt <- pl_lt %>%
  mutate(
    pop = factor(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 #Q: Does daily solar radiation positively affect daily growth rate across all populations?
#Test: Fit a mixed-effect model with population as a random slope.

```

pl_lt.lmer <- lmer(
  daily_growth ~ mean_light_ly_day2 +
    (1 + mean_light_ly_day2 | pop),
  data = pl_lt, REML = TRUE
)
summary(pl_lt.lmer)

## Linear mixed model fit by REML ['lmerMod']
## Formula: daily_growth ~ mean_light_ly_day2 + (1 + mean_light_ly_day2 |
##      pop)
##      Data: pl_lt
##
## REML criterion at convergence: 1656
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.1389 -0.2891 -0.0481  0.1374  8.6048
##
## Random effects:
##      Groups   Name                Variance Std.Dev. Corr
##      pop      (Intercept)          0.02193  0.14807
##              mean_light_ly_day2  0.00437  0.06611  0.97
##      Residual                    0.11366  0.33713
## Number of obs: 2415, groups:  pop, 23
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    0.13632   0.03902   3.494
## mean_light_ly_day2 0.07966   0.01999   3.986
##
## Correlation of Fixed Effects:
##              (Intr)
## mn_lght_l_2 0.916

```

overall effect of solar radiation on plant daily growth #Figure C: Relationship between daily average solar radiation and predicted daily growth rate (cm/day), aggregated across all populations. Each hexagon represents the density of observations. The black line shows the fitted regression slope .

```

#Unscaling
pred_all <- ggpredict(pl_lt.lmer, terms = "mean_light_ly_day2") %>%
  as.data.frame() %>%
  mutate(light_orig = x * sd_lt + mean_lt)

#Plot
p_overall <- ggplot() +
  geom_hex(data = pl_lt,
    aes(x = mean_light_ly_day, 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 = "Figure C: Effect of Daily Light on Daily Growth (overall)",
    x = "Daily Avg Light (ly/day)",
    y = "Predicted Daily Growth (cm/day)") +
  theme_bw() +
  theme(panel.grid.minor = element_blank())
p_overall

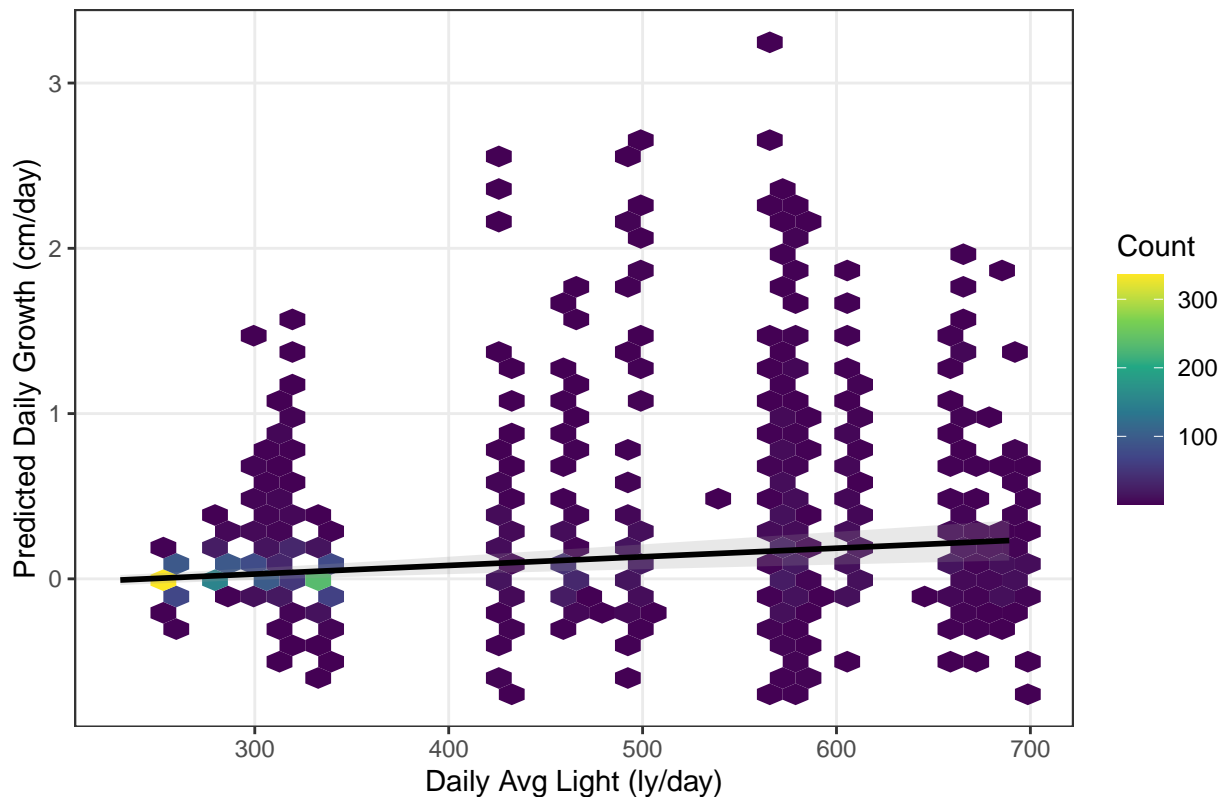
```

```

## Warning: Removed 11176 rows containing non-finite outside the scale range
## (`stat_binhex()`).

```

Figure C: Effect of Daily Light on Daily Growth (overall)



Effect of solar radiation on plant daily growth by population #Q: Do different populations vary in their growth response to daily solar radiation? #Test: For visualization, scatter plots with fitted regression lines were drawn separately for each population. #Figure D: Light–growth relationship by population. #Scatter plots show daily growth rate (cm/day) against daily average solar radiation (ly/day) for 22 populations. Each panel corresponds to one population, with the blue line indicating the fitted linear trend.

```
p_facet <- ggplot(pl_lt,
  aes(mean_light_ly_day, daily_growth)) +
  facet_wrap(~ pop, ncol = 6) +
  geom_point(alpha = .15, size = .6, color = "grey35") +
  geom_smooth(method = "lm", se = FALSE, linewidth = .8) +
  labs(title = "Figure D: Light-Growth relationship by population",
    x = "Daily Avg Light (ly/day)",
    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())
```

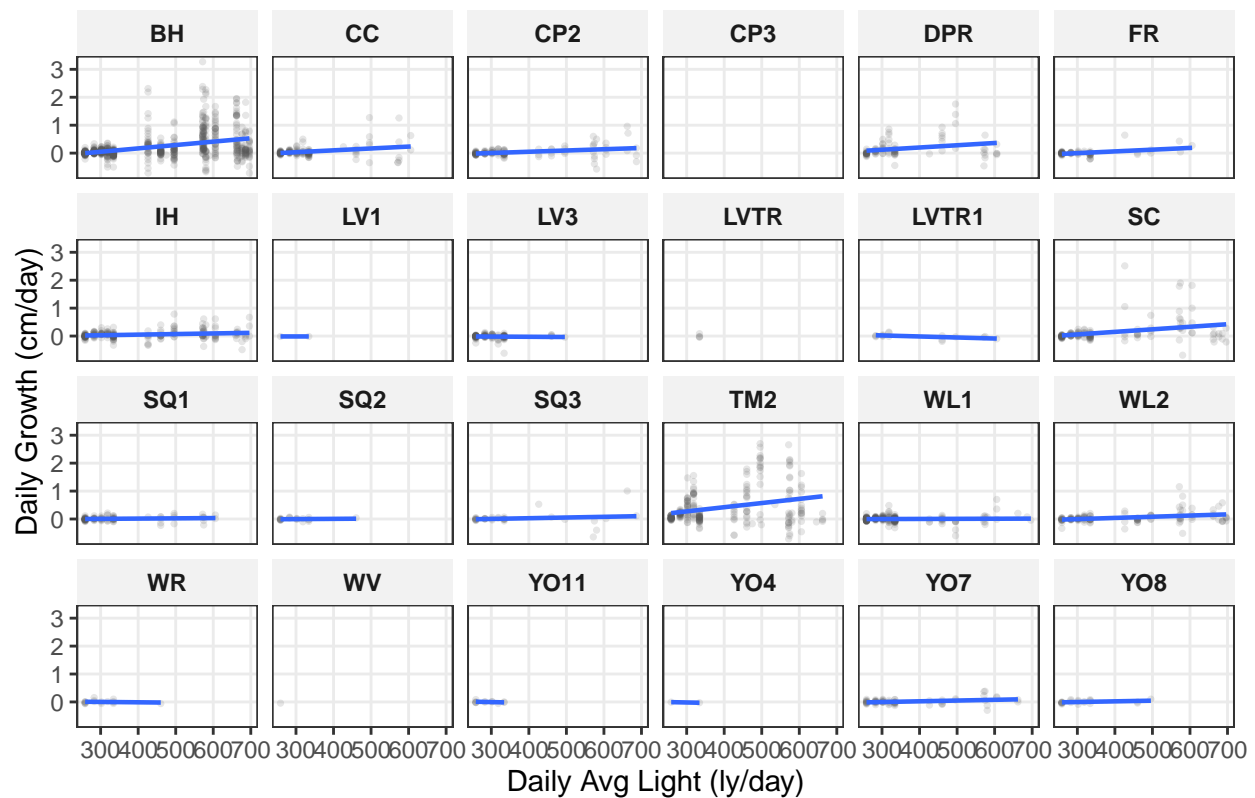
```
p_facet
```

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

```
## Warning: Removed 11176 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 11176 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Figure D: Light–Growth relationship by population



Find slope of plant growth and solar radiation for each population #Q: Which populations show significantly positive or negligible slopes in the light–growth relationship? #Method: Extract slopes and 95% confidence intervals for each population from the mixed-effects model #Figure E. Population-specific slopes with 95% confidence intervals. #The slope of the light–growth relationship (cm/day per Ly/day) is shown for each population. Points indicate estimated slopes and horizontal lines show 95% CIs. The vertical dashed line marks zero effect. Most populations exhibit positive slopes, suggesting that higher solar radiation is generally associated with faster daily growth, although the strength and significance of this relationship vary across populations.

```

#slope and standard deviation of fixed effects
b_fix <- fixef(pl_lt.lmer)["mean_light_ly_day2"]
V_fix <- vcov(pl_lt.lmer)["mean_light_ly_day2","mean_light_ly_day2"]

#slope and standard deviation of random effects
re <- ranef(pl_lt.lmer, condVar = TRUE)
re_pop <- re$pop
postVar <- attr(re$pop, "postVar")
sl_col <- which(colnames(re_pop) == "mean_light_ly_day2")

#create the tibble of slopes for each population
pop_slope <- tibble(
  parent_pop = rownames(re_pop),
  rand_slope = re_pop[, "mean_light_ly_day2"],
  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
  )

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

```

```

## # A tibble: 23 x 4
##   parent_pop slope_per_Wm2 lower_per_Wm2 upper_per_Wm2
##   <chr>          <dbl>          <dbl>          <dbl>
## 1 BH             0.00113        0.000837        0.00142
## 2 CC             0.000595        0.000124        0.00106
## 3 CP2            0.000425        0.0000137       0.000835
## 4 DPR            0.000889        0.000429        0.00135

```

```
## 5 FR          0.000462  -0.0000947  0.00102
## 6 IH          0.000307  -0.0000424  0.000656
## 7 LV1         0.000503  -0.000362   0.00137
## 8 LV3         0.000251  -0.000328   0.000830
## 9 LVTR        0.000481  -0.000347   0.00131
## 10 LVTR1      0.000208  -0.000393   0.000808
## # i 13 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 = "Figure E: Population-specific slopes with 95% CI",
       x = "Population", y = "Slope (cm/day per ly/day)") +
  theme_bw()
```

