Analysis of a clonal dataset

In this analysis, following the conclusions of the theoretical experiment, we aim at testing if intraspecific variability in observed individual growth can emerge from the environment. Therefore, we use a clonal experimental setup. The Eucflux dataset is a clonal trial on Eucalyptus varieties. One of its main goals is to determine the productivity of each clone. One hypothesis on intraspecific variability in growth is that it mainly derives from environmental factors, and not only from genetic, intrinsic factors. Therefore, we can use the Eucflux dataset in order to ask if we can we detect some intraspecific variability of growth within clones. As we suppose growth to be mainly the result of environmental variables, we expect to detect intraspecific variability of growth within clones (or single genotypes), although we also expect the genotype to influence the growth response.

The EUCFLUX setup is located in Brasilia, in the state of Sao Paulo. It has 16 genotypes (14 clones and two seed-origin genotypes) of 5 different species or hybrides, which grow on 10 sites and were monitored over 6 years. The detail of the setup is detailed in le Maire et al., 2019. We used the DBH measured during 5 complete censuses in order to compute annual growth in mm/year.

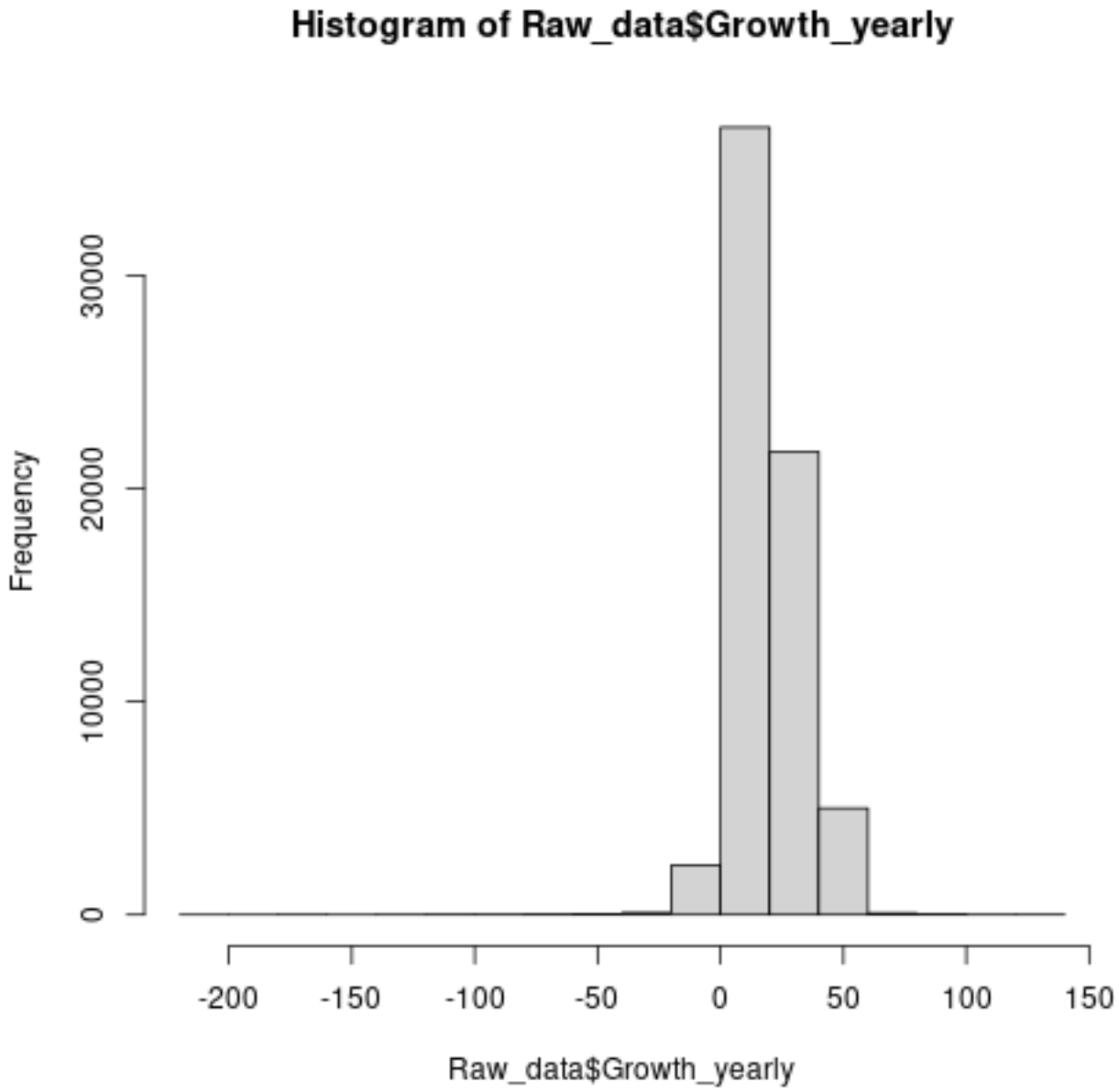
The raw data of the experiment was manually rearranged into six files with LibreOffice Calc, each file corresponding to a complete census. The seed-origin individuals are removed from the data since they contain genetic individual variability. We keep only the dates of measurements for which the code is "1" for complete DBH measure in Le Maire et al. 2019. We compute annualised growth in mm/y as the difference between two consecutive censuses divided by the time between the two censuses and we remove all negative growth values. We compute the neperian logarithm of diameter and of growth (with a constant for growth in order to avoid undefined values).

```
Full_Data <- list()
for (k in 1:6) {
    l = c(3, 4, 6, 8, 12, 14)[k]
    Full_Data[[k]] <- read.table(here::here("data", "EUCFLUX",
        paste0("Eucflux_Data_", l, ".csv")), header = T, sep = ",")
}

Raw_data <- do.call("rbind", Full_Data)

colnames(Raw_data) <- c("Site", "Gen", "Tree_number", "Date",
    "CBH")

# remove seed-origin individuals
Raw_data <- Raw_data[which(!(Raw_data$Gen %in% c(1, 2))), ]

Dates <- read.table(here::here("data", "EUCFLUX", "Dates.csv"),
    header = T, sep = ",")

# Date selection : only keep dates with code '1' for DBH in
# table 2 in https://doi.org/10.1016/j.foreco.2019.06.040 '1'
# = all blocks; all genotypes; all inside plot trees.
Dates_measurements <- c("01/06/2011", "01/01/2012", "15/01/2013",
    "15/02/2014", "15/02/2015", "15/01/2016")
```

```
Dates <- Dates[which(Dates$X %in% Dates_measurements), ]

# Calculating the interval in days between two censuses
Dates <- Dates %>%
    dplyr::rename(Date_number = DATES, Date_2 = X) %>%
    dplyr::mutate(Date_2 = stringr::str_replace_all(Date_2, "/",
        "-"), Date_1 = c(NA, Date_2[1:(nrow(Dates) - 1)])) %>%
    dplyr::mutate(Date_2 = lubridate::dmy(Date_2), Date_1 = lubridate::dmy(Date_1),
        Interval = as.integer(difftime(time1 = Date_2, time2 = Date_1)))

# Calculating the growth in m between two censuses, then
# convert it in mm/year
Raw_data <- Raw_data %>%
    dplyr::mutate(DBH = CBH/pi, Tree = paste0(Site, "_", Gen,
        "_", Tree_number)) %>%
    dplyr::arrange(Site, Gen, Tree_number, Date) %>%
    dplyr::mutate(Growth = rep(0, (nrow(Raw_data))), DBH = dplyr::na_if(DBH,
        0)) %>%
    dplyr::rename(D_2 = DBH) %>%
    # grouping is necessary in order not to mix the data of
# different trees
dplyr::group_by(Tree) %>%
    # n() is the number of rows within the group, i.e. per tree
dplyr::mutate(D_1 = c(NA, D_2[1:(dplyr::n() - 1)])) %>%
    dplyr::mutate(Growth = D_2 - D_1) %>%
    # growth in mm/y
dplyr::mutate(Growth_yearly = (Growth/(Dates$Interval/365)) *
    10) %>%
    dplyr::ungroup() %>%
    dplyr::mutate(Gen = as.factor(Gen), Site = as.factor(Site),
        Tree = as.factor(Tree), Date = as.factor(Date))

grDevices::png(here::here("outputs", "clonal_analysis", "figures",
    "Data_growth_raw.png"))
hist(Raw_data$Growth_yearly)
dev.off()
```

```
## pdf
##   2
```

```
knitr::include_graphics(here::here("outputs", "clonal_analysis",
    "figures", "Data_growth_raw.png"))
```




```
summary(Raw_data$Growth_yearly)
```

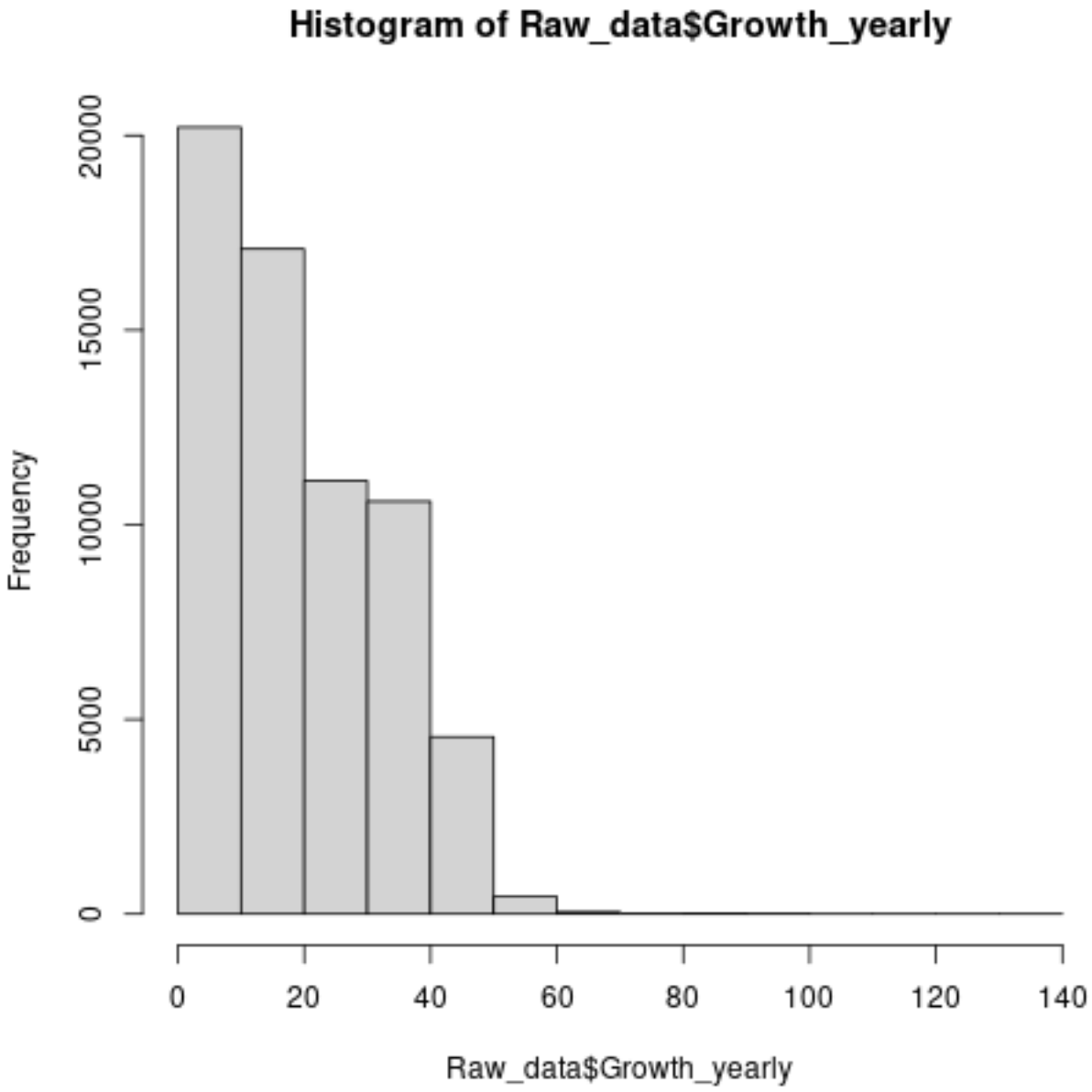
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## -206.849    7.321   15.438   18.320   29.352  133.372    17960
```

```
Raw_data <- Raw_data[which(Raw_data$Growth_yearly >= 0), ]
grDevices::png(here::here("outputs", "clonal_analysis", "figures",
    "Data_growth_filtered.png"))
hist(Raw_data$Growth_yearly)
dev.off()
```

```
## pdf
```

```
##   2
```

```
knitr::include_graphics(here::here("outputs", "clonal_analysis",
    "figures", "Data_growth_filtered.png"))
```



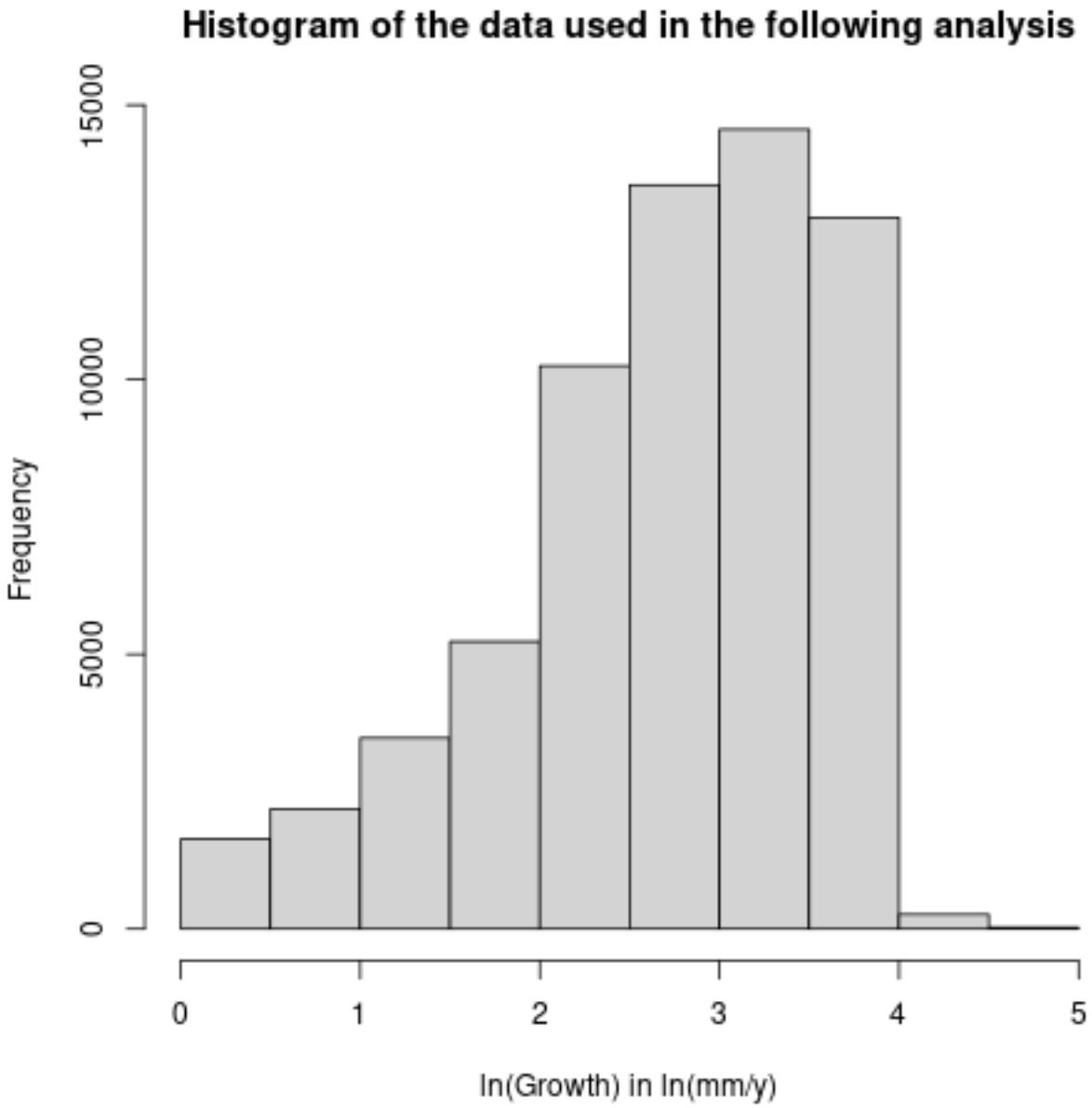

```
summary(Raw_data$Growth_yearly)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   8.117  16.137  19.139  29.860 133.372
```

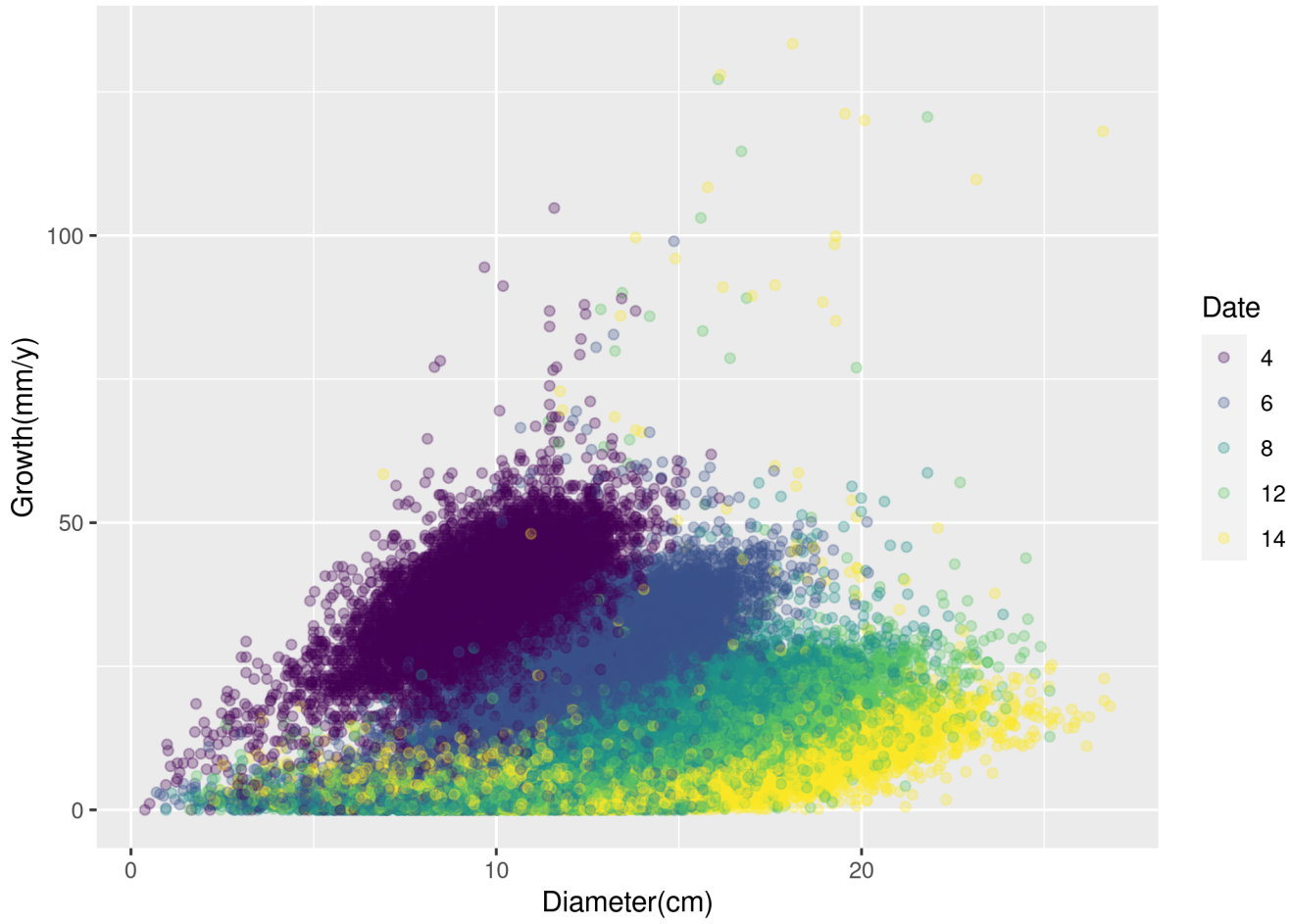
```r
# logging G and D in order to have normal data constant to
# avoid logging null values
Raw_data$logG <- log(Raw_data$Growth_yearly + 1)
Raw_data$logD <- log(Raw_data$D_1)

grDevices::png(here::here("outputs", "clonal_analysis", "figures",
    "Data_log_growth.png"))
hist(Raw_data$logG, xlab = "ln(Growth) in ln(mm/y)", main = "Histogram of the data used in the followin
dev.off()
```

```
## pdf
##   2
```

```r
knitr::include_graphics(here::here("outputs", "clonal_analysis",
    "figures", "Data_log_growth.png"))
```




```
g <- ggplot2::ggplot(Raw_data, ggplot2::aes(x = D_2, y = Growth_yearly,
    col = Date)) + ggplot2::geom_point(alpha = 0.3) + ggplot2::xlab("Diameter(cm)") +
    ggplot2::ylab("Growth(mm/y)") + ggplot2::scale_colour_viridis_d()
ggplot2::ggsave(filename = "Raw_data.png", plot = g, path = here::here("outputs",
    "clonal_analysis", "figures"), device = "png", width = 7,
    height = 5)
knitr::include_graphics(here::here("outputs", "clonal_analysis",
    "figures", "Raw_data.png"))
```




```
Data <- Raw_data
```

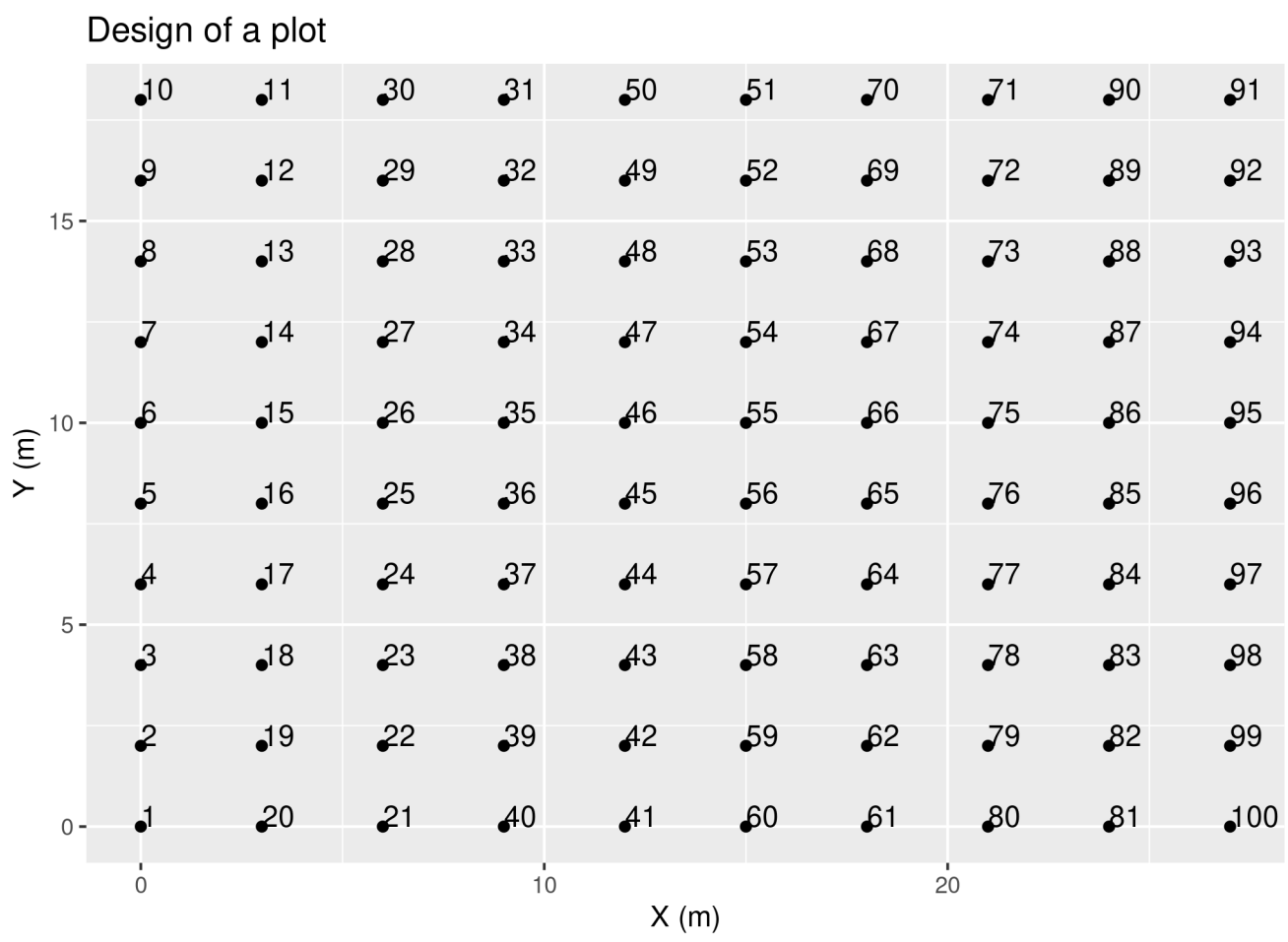
This figure shows that the date of measure has a big influence on the values of growth but also on the relationship between growth and diameter : the slope seems to be smaller with time, indicating that for the same diameter, growth is slower through time. This is likely an effect of competition for light and possibly underground resources, since as the trees grow their capacity to capture resources increases.

Therefore, we compute a competition index to integrate this effect in the growth model. The competition index is computed for each tree which is not on the edge of a plot. It is the sum of the basal areas of the direct neighbours divided by the area of the rectangle that comprises all the direct neighbours. It is then logged.

$C_{i,t} = \frac{\sum BA_{neighbours(i,t)}}{A}$, A is constant.

```
# We must retrieve the coordinates of each tree in order to
# show the design of plots

Coordinates <- data.frame(Tree_number = c(c(1:10), c(20:11),
    c(21:30), c(40:31), c(41:50), c(60:51), c(61:70), c(80:71),
    c(81:90), c(100:91)), X = rep(1:10, each = 10), Y = rep(1:10,
    10))
# convert in meters horizontally, two trees are 3 m apart and
# vertically 2 m apart.
```

```
Coordinates <- Coordinates %>%
    # Origin = 0,0 and conversion in meters
dplyr::mutate(X_m = (X - 1) * 3, Y_m = (Y - 1) * 2)

g <- ggplot2::ggplot(data = Coordinates, ggplot2::aes(x = X_m,
    y = Y_m)) + ggplot2::geom_point() + ggplot2::coord_fixed(ratio = 1) +
    ggplot2::geom_text(ggplot2::aes(label = Tree_number, hjust = 0,
        vjust = 0)) + ggplot2::labs(title = "Design of a plot") +
    ggplot2::xlab("X (m)") + ggplot2::ylab("Y (m)")
ggplot2::ggsave(filename = "Plot_design.png", plot = g, path = here::here("outputs",
    "clonal_analysis", "figures"), device = "png", width = 7,
    height = 5)
knitr::include_graphics(here::here("outputs", "clonal_analysis",
    "figures", "Plot_design.png"))
```




```
# Competition index = sum of basal area of direct neighbours
# / area of the rectangle Trees are 2 m apart in vertically
# and 3 m apart horizontally, so the area of the rectangle is
# of 4*6 m = 24m^2
A = 24

# Computing basal area of each tree at each date
```

```r
Data$BA <- pi * ((Data$D_1/100/2)^2)

# First, we do not compute this index for trees at the edge

trees_not_edge <- c(1:100)[!(c(1:100) %in% c(1:11, 20, 21, 30,
    31, 40, 41, 50, 51, 60, 61, 70, 71, 80, 81, 90:100))]
# To know which trees are neighbour, we need the disposition
# of the trees in each plot
disposition <- matrix(c(c(10:1), c(11:20), c(30:21), c(31:40),
    c(50:41), c(51:60), c(70:61), c(71:80), c(90:81), c(91:100)),
    ncol = 10, nrow = 10)

Data$C <- numeric(nrow(Data))

for (k in 1:nrow(Data)) {
    if (Data[k, ]$Tree_number %in% trees_not_edge) {
        # Tree number
        tree <- Data[k, ]$Tree_number
        site <- Data[k, ]$Site
        gen <- Data[k, ]$Gen
        date <- Data[k, ]$Date
        # Position in the disposition matrix
        position <- which(disposition == tree, arr.ind = TRUE)
        # Numbers of the neighbours
        neighbours <- c(disposition[position[1] + 1, position[2] +
            1], disposition[position[1] - 1, position[2] - 1],
            disposition[position[1] - 1, position[2] + 1], disposition[position[1] +
                1, position[2] - 1], disposition[position[1],
                position[2] + 1], disposition[position[1] + 1,
                position[2]], disposition[position[1], position[2] -
                1], disposition[position[1] - 1, position[2]])
        neighbours <- paste0(site, "_", gen, "_", neighbours)
        sum_BA <- sum(Data[which(Data$Tree %in% neighbours &
            Data$Date == date), ]$BA)
        Data[k, ]$C <- (sum_BA/A) * 1000
    }
}

Data$logC <- log(Data$C)
Data_compet <- Data[which((Data$Tree_number %in% trees_not_edge)),
    ]

save(Data_compet, file = here::here("outputs", "clonal_analysis",
    "Data_compet.RData"))
```

In order to partition the variance of individual growth data, we built a hierarchical Bayesian model and used Stan and the packages rstan and brms to implement it in R. We used the following parameters : n.adapt = 1000 ; n.burn = 1000 ; n.iter = 5000 ; n.thin = 5.

Our model incorporated a fixed effect on the intercept (β_0), on the slop of diameter D (β_1),and on the competition index C (β_2) and several random effects, namely temporal (date of census, b_t), individual (tree identifier, b_i), spatial (site, b_s), and genotype (b_g).

$$ln(G_{i,t} + 1) = (\beta_0 + b_i + b_s + b_g + b_t) + \beta_1 \times ln(D_{i,t}) + \beta_2 \times ln(C_{i,t}) + \epsilon_{i,t}$$

Priors

$\beta_0 \sim \mathcal{N}(mean = 0, var = 1), iid$

$\beta_1 \sim \mathcal{N}(mean = 0, var = 1), iid$

$\beta_2 \sim \mathcal{N}(mean = 0, var = 1), iid$

$b_i \sim \mathcal{N}(mean = 0, var = V_i), iid$

$b_s \sim \mathcal{N}(mean = 0, var = V_s), iid$

$b_g \sim \mathcal{N}(mean = 0, var = V_g), iid$

$b_t \sim \mathcal{N}(mean = 0, var = V_t), iid$

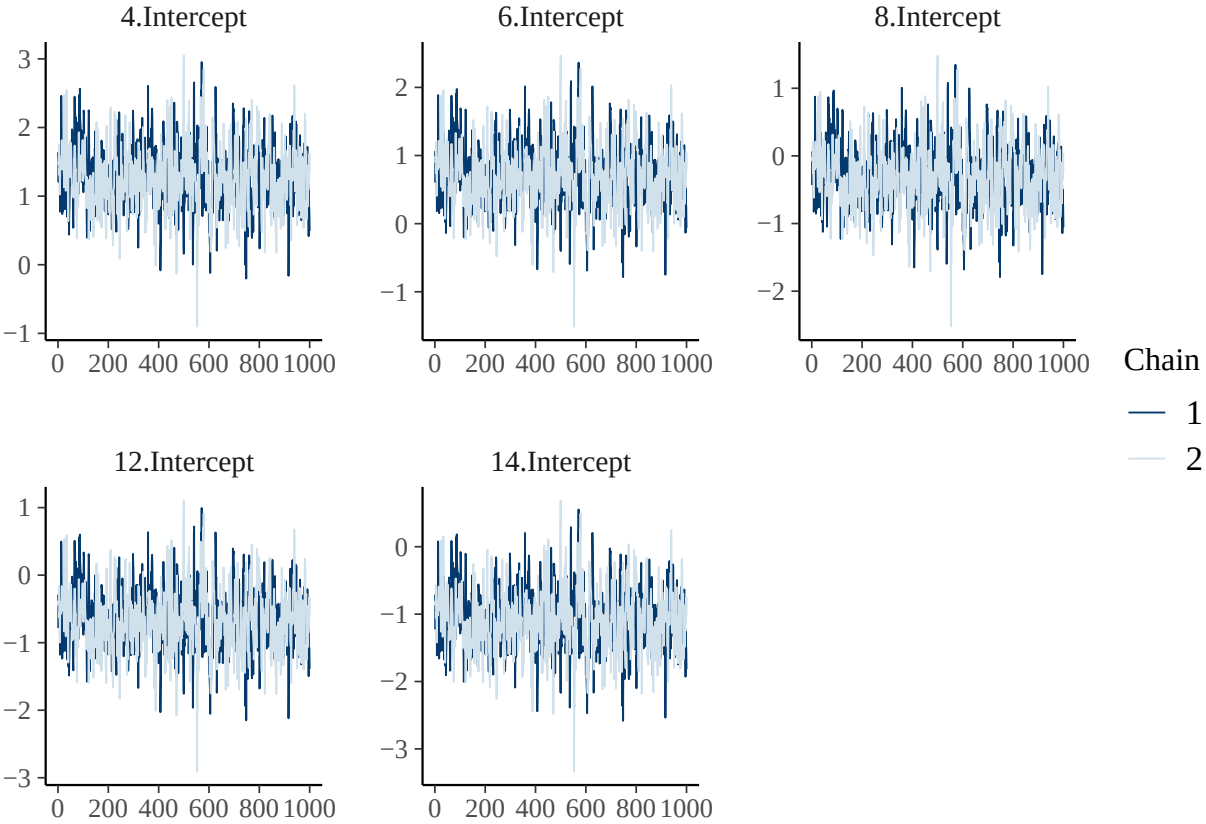
$\epsilon_{i,t} \sim \mathcal{N}(mean = 0, var = V), iid$

Hyperpriors

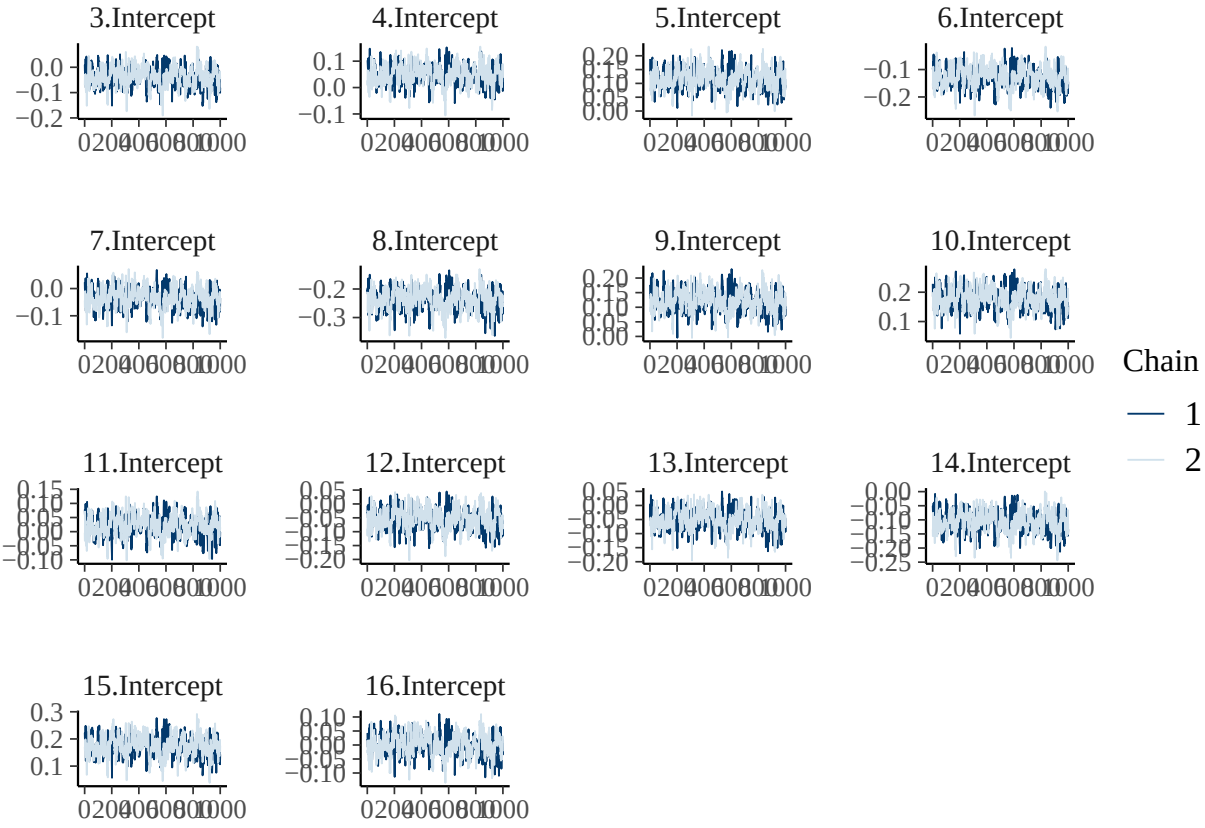
$V_i \sim \mathcal{IG}(shape = 10^{-3}, rate = 10^{-3}), iid$

After convergence of the model, we examined the variance of each random effect, and this enabled us to perform a variance partition.

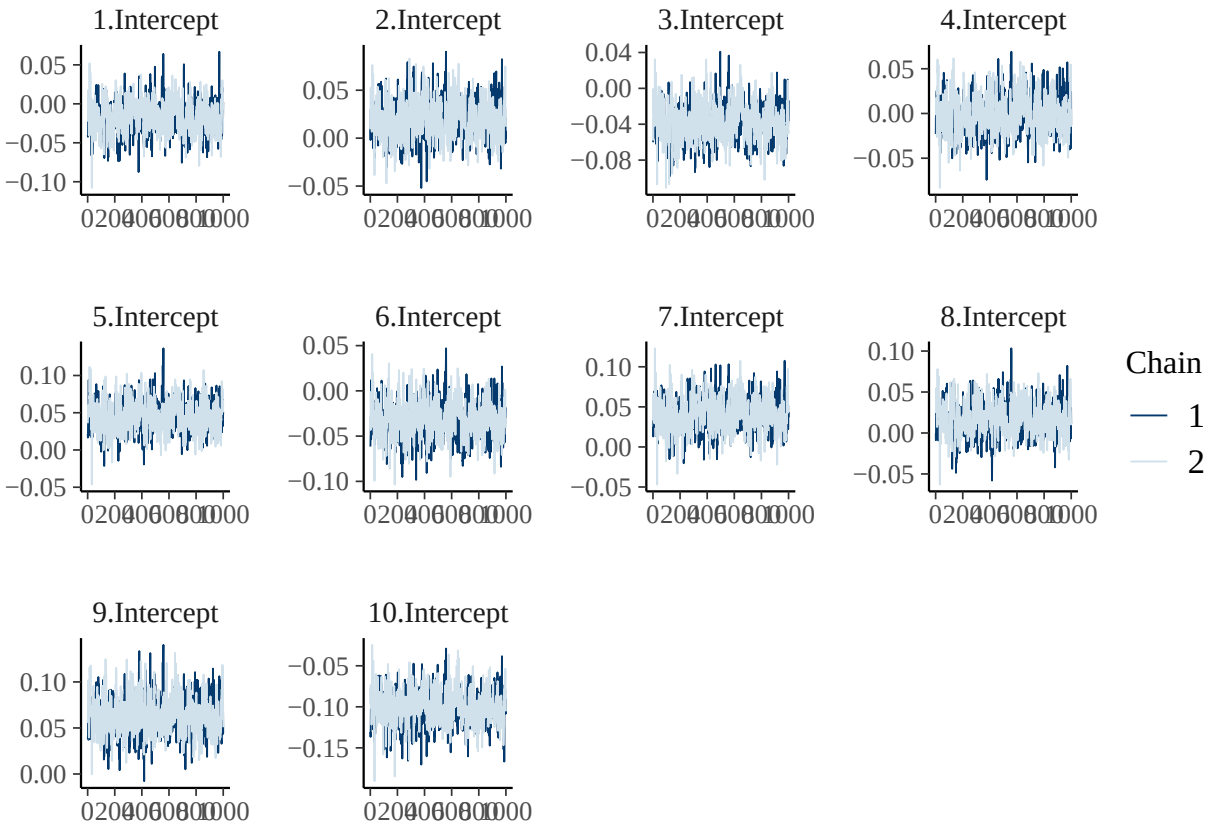
```
load(here::here("outputs", "clonal_analysis", "brms_mod.RData"))

# Check convergence of random effects et some features of the
# random effects
Ranef_date <- as.data.frame(brms::ranef(brms_mod, summary = F)$Date)
Ranef_date$Chain <- c(rep(1, 1000), rep(2, 1000))
bayesplot::mcmc_trace(Ranef_date)
```



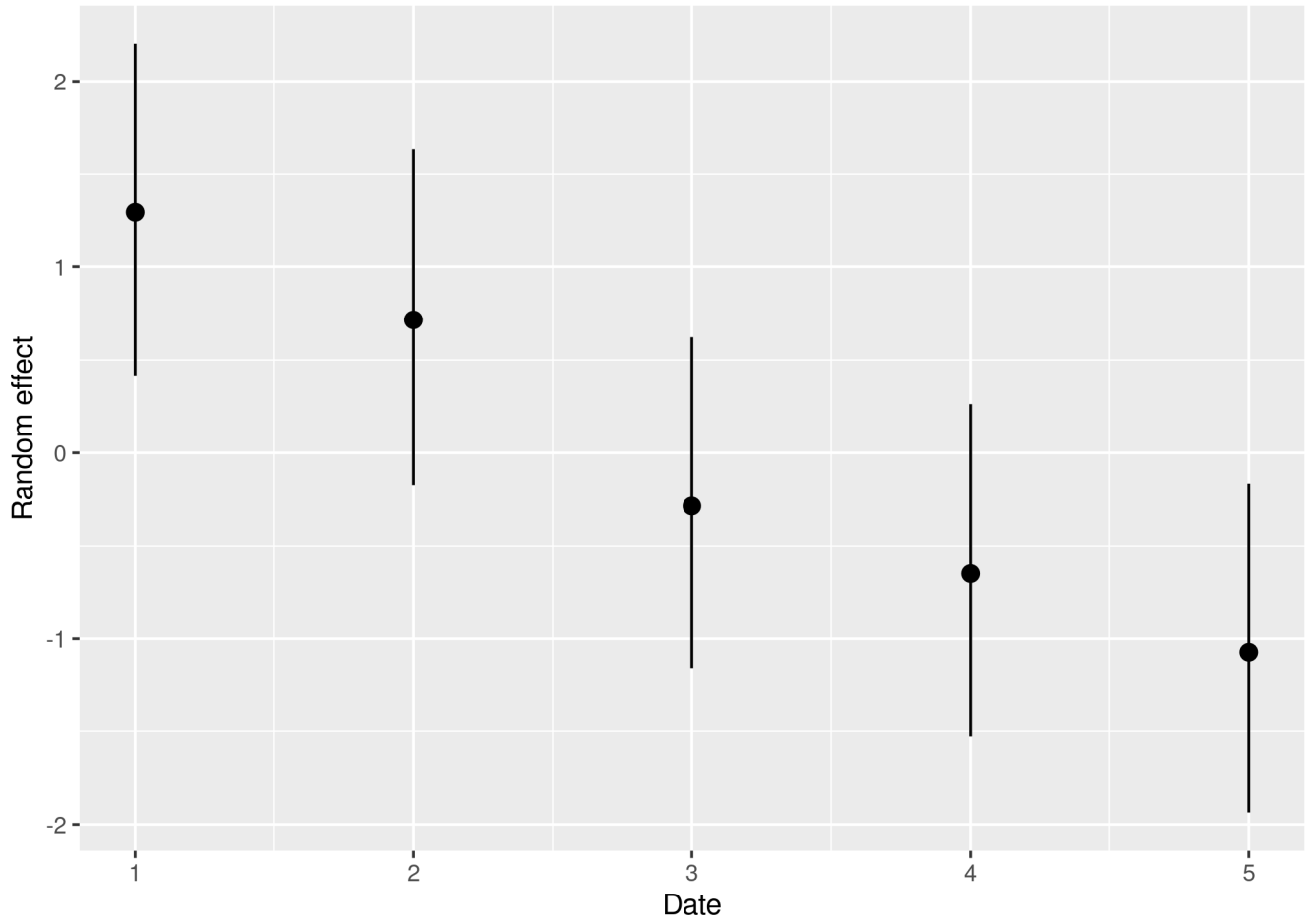
```
Ranef_date$Date <- rownames(Ranef_date)

Ranef_gen <- as.data.frame(brms::ranef(brms_mod, summary = F)$Gen)
Ranef_gen$Chain <- c(rep(1, 1000), rep(2, 1000))
bayesplot::mcmc_trace(Ranef_gen)
```



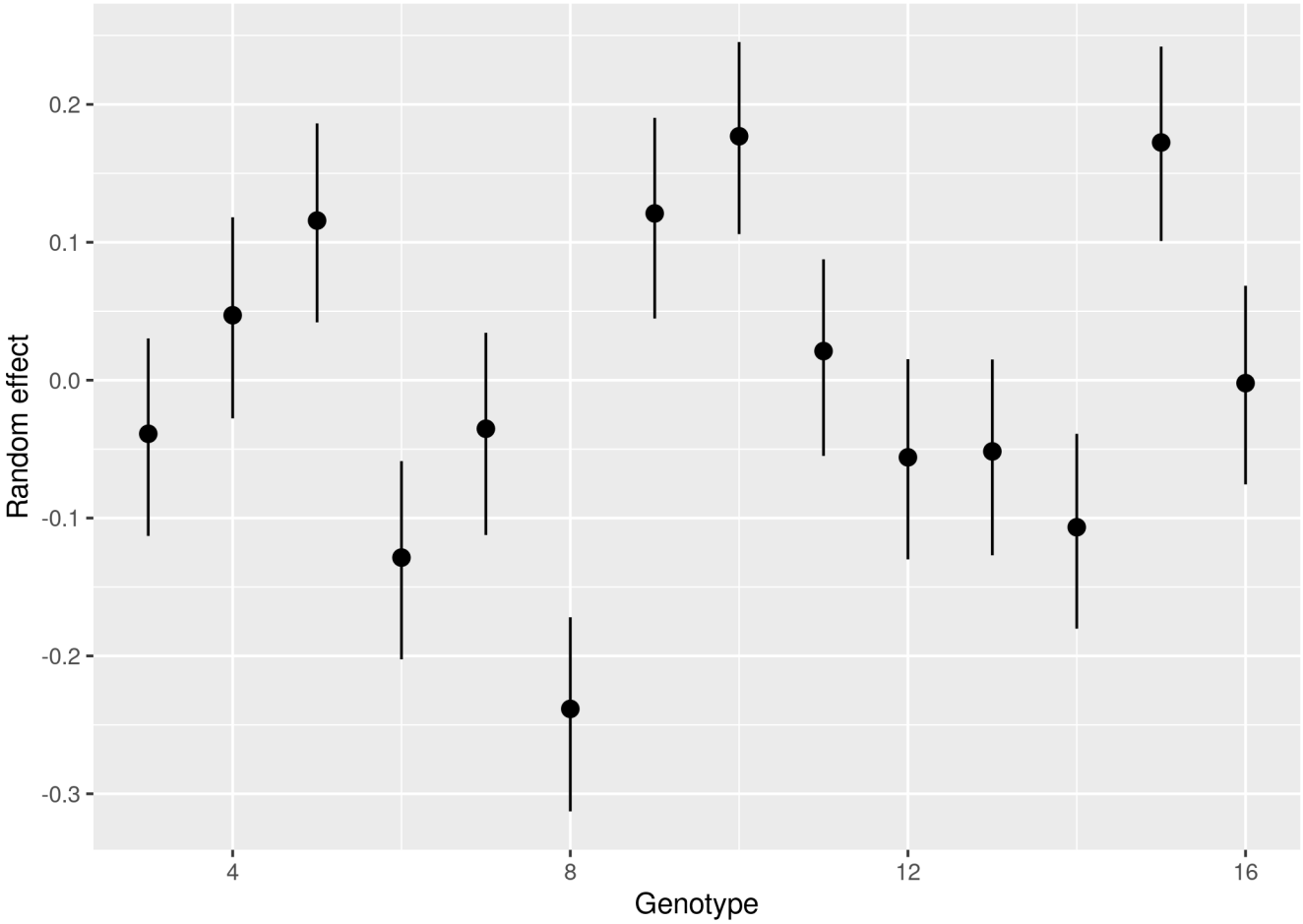

```
Ranef_site <- as.data.frame(brms::ranef(brms_mod, summary = F)$Site)
Ranef_site$Chain <- c(rep(1, 1000), rep(2, 1000))
bayesplot::mcmc_trace(Ranef_site)
```



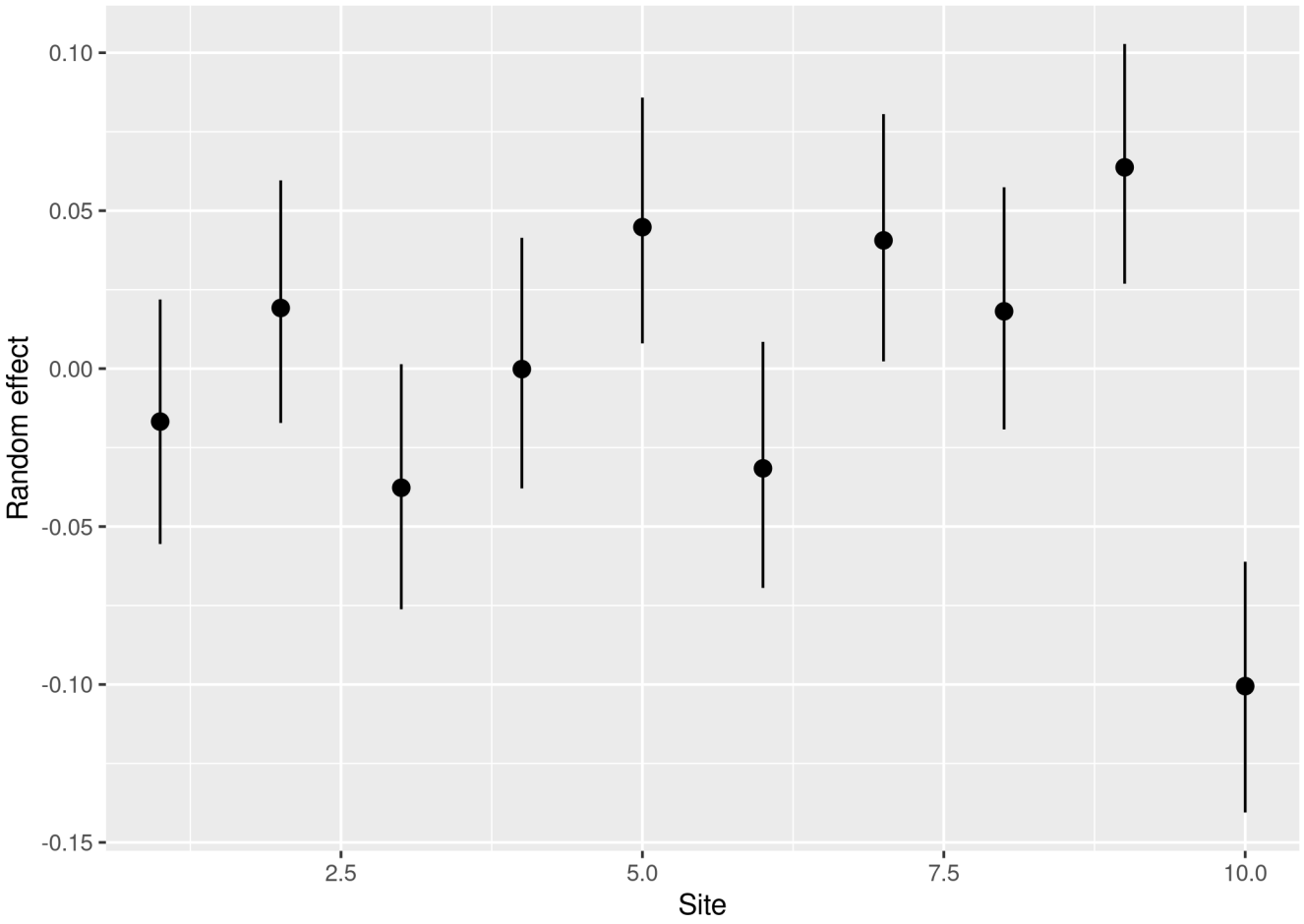

```
Ranef_date_summary <- as.data.frame(brms::ranef(brms_mod)$Date)
Ranef_date_summary$Date_nb <- c(1:nrow(Ranef_date_summary))
colnames(Ranef_date_summary) <- gsub(".", "_", colnames(Ranef_date_summary),
    fixed = T)

# Look at random effects values
g <- ggplot2::ggplot(data = Ranef_date_summary, ggplot2::aes(x = Date_nb,
    y = Estimate_Intercept)) + ggplot2::geom_point() + ggplot2::geom_pointrange(ggplot2::aes(ymin = Q2_
    ymax = Q97_5_Intercept)) + ggplot2::xlab("Date") + ggplot2::ylab("Random effect")
ggplot2::ggsave(filename = "Ranef_date.png", plot = g, path = here::here("outputs",
    "clonal_analysis", "figures"), device = "png", width = 7,
    height = 5)
knitr::include_graphics(here::here("outputs", "clonal_analysis",
    "figures", "Ranef_date.png"))
```




```
Ranef_Gen_summary <- as.data.frame(brms::ranef(brms_mod)$Gen)
Ranef_Gen_summary$Gen <- as.numeric(rownames(Ranef_Gen_summary))
colnames(Ranef_Gen_summary) <- gsub(".", "_", colnames(Ranef_Gen_summary),
    fixed = T)

g <- ggplot2::ggplot(data = Ranef_Gen_summary, ggplot2::aes(x = Gen,
    y = Estimate_Intercept)) + ggplot2::geom_point() + ggplot2::geom_pointrange(ggplot2::aes(ymin = Q2_
    ymax = Q97_5_Intercept)) + ggplot2::xlab("Genotype") + ggplot2::ylab("Random effect")
ggplot2::ggsave(filename = "Ranef_genotype.png", plot = g, path = here::here("outputs",
    "clonal_analysis", "figures"), device = "png", width = 7,
    height = 5)
knitr::include_graphics(here::here("outputs", "clonal_analysis",
    "figures", "Ranef_genotype.png"))
```



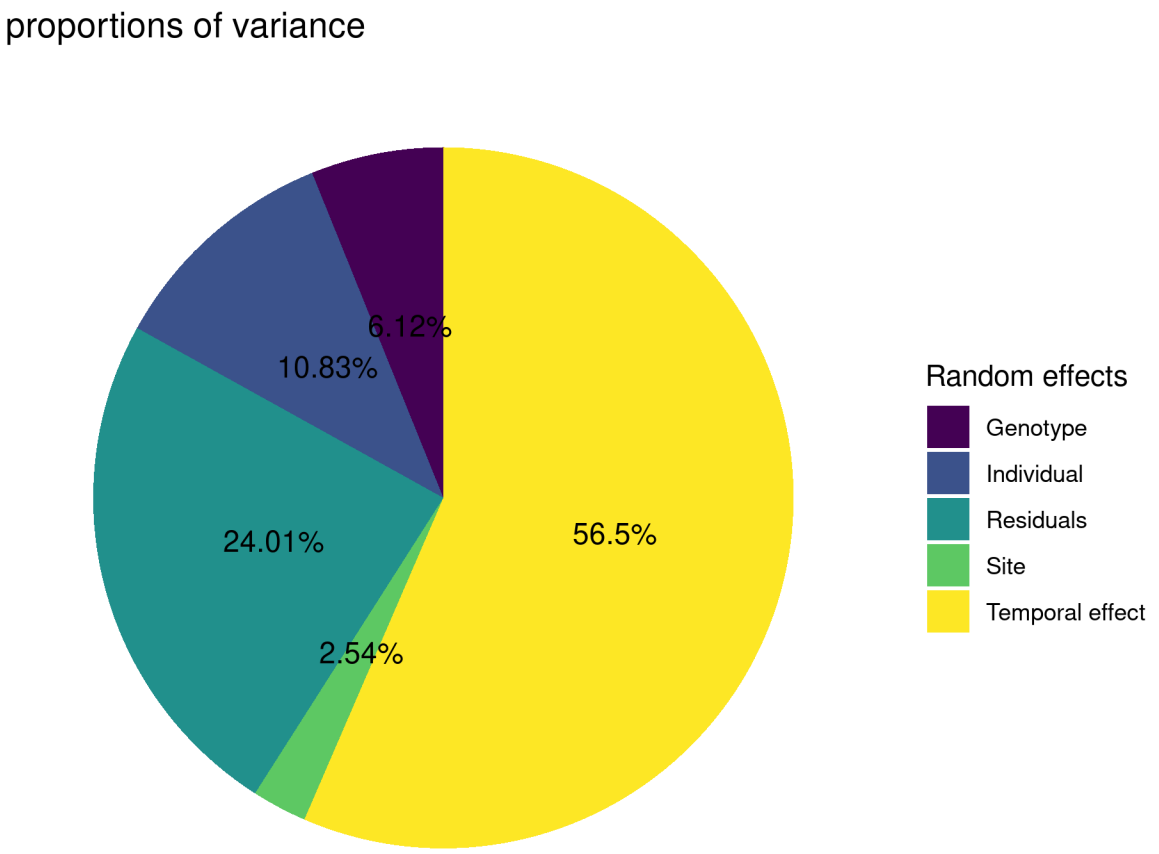

```
Ranef_Site_summary <- as.data.frame(brms::ranef(brms_mod)$Site)
Ranef_Site_summary$Site <- as.numeric(rownames(Ranef_Site_summary))
colnames(Ranef_Site_summary) <- gsub(".", "_", colnames(Ranef_Site_summary),
    fixed = T)

g <- ggplot2::ggplot(data = Ranef_Site_summary, ggplot2::aes(x = Site,
    y = Estimate_Intercept)) + ggplot2::geom_point() + ggplot2::geom_pointrange(ggplot2::aes(ymin = Q2_
    ymax = Q97_5_Intercept)) + ggplot2::xlab("Site") + ggplot2::ylab("Random effect")
ggplot2::ggsave(filename = "Ranef_site.png", plot = g, path = here::here("outputs",
    "clonal_analysis", "figures"), device = "png", width = 7,
    height = 5)
knitr::include_graphics(here::here("outputs", "clonal_analysis",
    "figures", "Ranef_site.png"))
```




```
Sum_EUCFLUX <- summary(brms_mod)
```

	Intercept (β_0)	Diameter (β_1)	Competition (β_2)	Individual variance (V_i)	Site variance (V_s)	Genetic variance (V_g)	Temporal variance (V_t)	Residuals variance (V)
Estimate	-3.6e-02	5.5e-01	-2.7e-01	2.3e-01	5.4e-02	1.3e-01	1.2e+00	5.1e-01
Estimation error	4.5e-01	5.1e-03	9.2e-03	4.1e-03	1.5e-02	2.8e-02	4.9e-01	2.1e-03

We obtained a good model convergence, and found that the two most important contributors to variance (apart from the residuals) were the date and individual identity. The effect of the genotype is quite small, and the effect of site is even smaller. The temporal random effects declined with the date, showing that the effect of the date on growth is negative (the older the trees become, the less they can grow). Therefore, the competition index C did not fully capture the effect of competition on growth.



The highest part of variance is explained by the census date. This is due to the negative effect of competition for light on growth, which increases with the growth of the trees. This means that our competition index does not completely capture competition.

The next random effect that explains most variance (apart from residuals) is the individual effect. This shows that there is individual variability even if the trees are clones and therefore that individual variability can be due to exogenous causes. This individual effect can be due to the micro-environment where it is positioned, but also to some individual history, here seedling manipulation and plantation.

The genotype explained 6.1% of the variance, which is surprising as some striking differences were denoted between clones beforehand. Therefore the impact of individual identity on growth is stronger than the effect of genotype. We did not find any relevant explanation for the mean random effect of each genotype.

Finally, the site has the littlest impact with 2.5% of the variance explained, this means that the physical environment between sites is quite homogeneous, or that the physical environment does not play a big role in growth. We know that at least three sites stand out : two for having clayey soil (8 and 9), and one for having been struck by lightning (10), leading to high mortality. Looking at the mean random site effects, we can see that the site 10 indeed has a lower random effect, and that sites 8 and 9 have a positive random effect. However, site 8 does not really depart from other sites. We can detect some site effects, but it is not always clear.

Difficulties of estimation for the intercept and the date random effect must be noted.

Overall, there is intraspecific variability within a clone. This shows that the environmental factors (in a broad sense : not genetic) have an impact on growth and that intraspecific variability can indeed emerge within a clone.