

# Report with embedded R code

Alyssa Reynolds, Zhijun Li, Runzi Qi

2022-03-22

GitHub Link: [https://github.com/zazauwu/432\\_Group5](https://github.com/zazauwu/432_Group5)

## Load Package

```
library(ape)
library(tidyr)
library(MASS)
library(MuMIN)
library(ggplot2)
library(reshape2)

##
## Attaching package: 'reshape2'
## The following object is masked from 'package:tidyr':
##   smiths

library(grid)
library(gridExtra)
library(ggfortify)
library(vegan)

## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.5-7
library(ggtree)

## ggtree v3.2.1 For help: https://yulab-smu.top/treedata-book/
##
## If you use ggtree in published research, please cite the most appropriate paper(s):
## 
## 1. Guangchuang Yu. Using ggtree to visualize data on tree-like structures. Current Protocols in Bioinformatics, 2018, 2018:1-14.
## 2. Guangchuang Yu, Tommy Tsan-Yuk Lam, Huachen Zhu, Yi Guan. Two methods for mapping and visualizing phylogenetic trees. Bioinformatics, 2015, 31(12):1933-1939.
## 3. Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam. ggtree: an R package for visualizing phylogenetic trees. Bioinformatics, 2020, 36(18):5022-5024.

##
## Attaching package: 'ggtree'
## The following object is masked from 'package:tidyr':
##   expand
```

```

## The following object is masked from 'package:ape':
##
##      rotate
library(dplyr)

##
## Attaching package: 'dplyr'
## The following object is masked from 'package:gridExtra':
##
##      combine
## The following object is masked from 'package:MASS':
##
##      select
## The following objects are masked from 'package:stats':
##
##      filter, lag
## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union
library(cowplot)

```

## Import Data

```

####Climate by site data
clim <- read.csv("./Data/HargreavesEcolLett data hobo&ibutton summary by site.csv")

####NK & HB lifetime reproductive success data
ltrsHB <- read.csv("./Data/HargreavesEcolLett data ltrsHB.csv")
ltrsNK <- read.csv("./Data/HargreavesEcolLett data ltrsNK.csv")

####NK & HB Performance data per subplot
proHB <- read.csv("./Data/HargreavesEcolLett data prop&LTFHB.csv")
proNK <- read.csv("./Data/HargreavesEcolLett data prop&LTFNK.csv")

```

## Climate Effect Analysis

### Data Check and Modification

```

sapply(clim, class)

##          transect           site d.snowpack.10.11      wTmin.10.11
##      "character"      "character"      "integer"        "numeric"
## Dec.FebTav.10.11      d.grows.11     GDD.grows.11      GDD.yr.11
##      "numeric"       "integer"       "integer"        "integer"
## avmaxminTJuly.11      avTJuly.11    d.snowpack.11.12      wTmin.11.12
##      "numeric"       "numeric"       "character"      "character"
## Dec.FebTav.11.12      d.grows.12     GDD.grows.12      GDD.yr.12
##      "character"      "character"      "character"      "character"
## avmaxminTJuly.12      avTJuly.12    d.snowpack.12.13      wTmin.12.13

```

```

##      "character"      "character"      "integer"      "numeric"
## Dec.FebTav.12.13 d.grows.13    GDD.grows.13    GDD.yr.13
##      "numeric"       "integer"       "character"     "character"
## avmaxminTJuly.13 avTJuly.13   d.snowpack.13.14 wTmin.13.14
##      "character"      "character"      "character"     "character"
## Dec.FebTav.13.14
##      "character"

```

#### Description of the variables:

**d.snowpack.10.11** = days with snowpack winter 2010-2011

**wTmin.10.11** = minimum temperature winter (C) at plant height 2010-2011

**Dec.FebTav.10.11** = mean temperature at plant height from Dec1st 2010 - Feb 28th 2011

**d.grows.11** = # days in growing season 2011 (start / end of gorwing season determined by sustained snow or frost)

**GDD.grows.11** = # growing degree days (Tbase=10C, Tmax capped at 30C) in growing season of 2011

**GDD.yr.11** = # growing degree days in 2011 (ie not truncated to estimated growing season)

**avmaxminTJuly.11** = mean July 2011 temperature (C) calculated from daily min and max T:  
mean(July1stmin + July1stmax + July2ndmin + July2ndmax + ... July31stmi + July31stmax)

**avTJuly.11** = mean July 2011 temperature (C) calculated from raw data (ie >2 measurements per day)

*#hence, convert the above columns into numeric variables*

```

cols.num <- c("d.snowpack.11.12", "wTmin.11.12", "Dec.FebTav.11.12", "d.grows.12", "GDD.grows.12", "GDD.yr.12")
clim[cols.num] <- sapply(clim[cols.num], as.numeric)

```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
sapply(clim, class)

##          transect      site d.snowpack.10.11      wTmin.10.11
## "character" "character"     "integer"      "numeric"
## Dec.FebTav.10.11      d.grows.11    GDD.grows.11      GDD.yr.11
## "numeric"     "integer"     "integer"      "integer"
## avmaxminTJuly.11    avTJuly.11    d.snowpack.11.12      wTmin.11.12
## "numeric"     "numeric"     "numeric"      "numeric"
## Dec.FebTav.11.12      d.grows.12    GDD.grows.12      GDD.yr.12
## "numeric"     "numeric"     "numeric"      "numeric"
## avmaxminTJuly.12    avTJuly.12    d.snowpack.12.13      wTmin.12.13
## "numeric"     "numeric"     "integer"      "numeric"
## Dec.FebTav.12.13      d.grows.13    GDD.grows.13      GDD.yr.13
## "numeric"     "integer"     "numeric"      "numeric"
## avmaxminTJuly.13    avTJuly.13    d.snowpack.13.14      wTmin.13.14
## "numeric"     "numeric"     "numeric"      "numeric"
## Dec.FebTav.13.14      "numeric"      "numeric"      "numeric"

head(clim)

##   transect site d.snowpack.10.11 wTmin.10.11 Dec.FebTav.10.11 d.grows.11
## 1      NK    L        182      -8.6       -2.0      121
## 2      NK    M        143     -11.5      -5.3      134
## 3      NK    E        103     -15.0      -8.7      124
## 4      NK   A1        229      -5.2      -1.8      55
## 5      NK   A2        87     -14.0      -6.6      67
## 6      HB    L        151      -2.6      -0.9      188
##   GDD.grows.11 GDD.yr.11 avmaxminTJuly.11 avTJuly.11 d.snowpack.11.12
## 1        975    1117        18.4      15.8      162
## 2        934    936         19.1      16.8      NA
## 3        762    762        15.6      12.9      120
## 4        396    480        10.9       8.2      279
## 5        361    443        11.0       9.2      88
## 6       1079   1083        18.7      17.3      110
##   wTmin.11.12 Dec.FebTav.11.12 d.grows.12 GDD.grows.12 GDD.yr.12
## 1      -12.90      -2.5      138     1006     1174
## 2       -8.15       NA       NA       NA       NA
## 3      -13.40      -7.6      107      704      808
## 4      -4.70      -7.6       67       NA       NA
## 5     -16.10      -7.6       68      386      513
## 6      -9.50      -2.8      197     1156     1164
##   avmaxminTJuly.12 avTJuly.12 d.snowpack.12.13 wTmin.12.13 Dec.FebTav.12.13
## 1        19.4      17.4      188      -7.4      -1.6
## 2        16.1      14.6      145      -8.0      -5.4
## 3        17.0      14.9      140      -8.5      -5.4
## 4         NA       NA      256      -2.0      -0.7
## 5        13.1      12.0      154      -8.6      -5.1
## 6        19.0      18.1      107     -12.6     -4.6
##   d.grows.13 GDD.grows.13 GDD.yr.13 avmaxminTJuly.13 avTJuly.13
## 1        137       NA       NA       NA       NA
## 2        159      760      760      16.1     14.70

```

```

## 3      123      601      604      14.3    13.20
## 4      91       409      409      11.7    10.00
## 5      92       370      381      11.3    9.40
## 6     186      1134     1613     18.2    17.16
##   d.snowpack.13.14 wTmin.13.14 Dec.FebTav.13.14
## 1      186      -8.0     -1.20
## 2      196      -8.5     -2.90
## 3      128     -20.0     -8.20
## 4      283      -2.5     -0.10
## 5       NA       NA       NA
## 6     142      -0.5      0.18

#combine 'transect' and 'site' info as a unique ID for each row
clim$ID <- as.factor(paste(clim$transect, clim$site, sep='.'))

row.names(clim) <- clim$ID

#remove 'transect' and 'site'
drops <- c("transect", "site")
clim <- clim[ , !(names(clim) %in% drops)]

#move 'ID' to the first column
clim <- clim %>%
  relocate(ID)

#use 'melt' to convert the 'clim' data.frame into long format in which 'ID' is identifier variable, whi
clim.me <- melt(clim, id.vars = "ID", measure.vars = 2:28)
head(clim.me)

##           ID      variable value
## 1  NK.L d.snowpack.10.11   182
## 2  NK.M d.snowpack.10.11   143
## 3  NK.E d.snowpack.10.11   103
## 4 NK.A1 d.snowpack.10.11   229
## 5 NK.A2 d.snowpack.10.11    87
## 6 HB.L d.snowpack.10.11   151

#use regular expression 'grepl' and 'gsub' to add 'year' column and remove date label in the 'variable'
clim.me$year <- ifelse(grepl('10.11', clim.me$variable),
  'winter 10-11',
  ifelse(grepl('11.12', clim.me$variable),
    'winter 11-12',
    ifelse(grepl('12.13', clim.me$variable),
      'winter 12-13',
      ifelse(grepl('13.14', clim.me$variable),
        'winter 13-14',
        ifelse(grepl('grows.11|July.11|yr.11', clim.me$variable),
          'year 2011',
          ifelse(grepl('grows.12|July.12|yr.12', clim.me$variable),
            'year 2012',
            ifelse(grepl('grows.13|July.13|yr.13', clim.me$variable),
              'year 2013',
              NA))))))

clim.me <- clim.me %>%
  mutate(variable = gsub(".(\d).*", "\2", variable))

```

```

head(clim.me)

##      ID  variable value      year
## 1  NK.L d.snowpack   182 winter 10-11
## 2  NK.M d.snowpack   143 winter 10-11
## 3  NK.E d.snowpack   103 winter 10-11
## 4  NK.A1 d.snowpack   229 winter 10-11
## 5  NK.A2 d.snowpack    87 winter 10-11
## 6  HB.L d.snowpack   151 winter 10-11

#use 'dcast' to cast the molten data.frame into a data frame
clim.ca <- dcast(clim.me, ID + year ~ variable)
clim.ca$year <- as.factor(clim.ca$year)
head(clim.ca)

##      ID      year avmaxminTJuly avTJuly d.grows d.snowpack Dec.FebTav
## 1  HB.A winter 10-11          NA       NA       NA      25     -11.6
## 2  HB.A winter 11-12          NA       NA       NA      NA       NA
## 3  HB.A winter 12-13          NA       NA       NA      51     -11.3
## 4  HB.A winter 13-14          NA       NA       NA      60     -10.0
## 5  HB.A     year 2011        14.0     10.7      91      NA       NA
## 6  HB.A     year 2012        12.4     11.4      NA      NA       NA

##      GDD.grows GDD.yr wTmin
## 1           NA     NA -22.3
## 2           NA     NA     NA
## 3           NA     NA -17.7
## 4           NA     NA -17.7
## 5          612    697     NA
## 6          378    461     NA

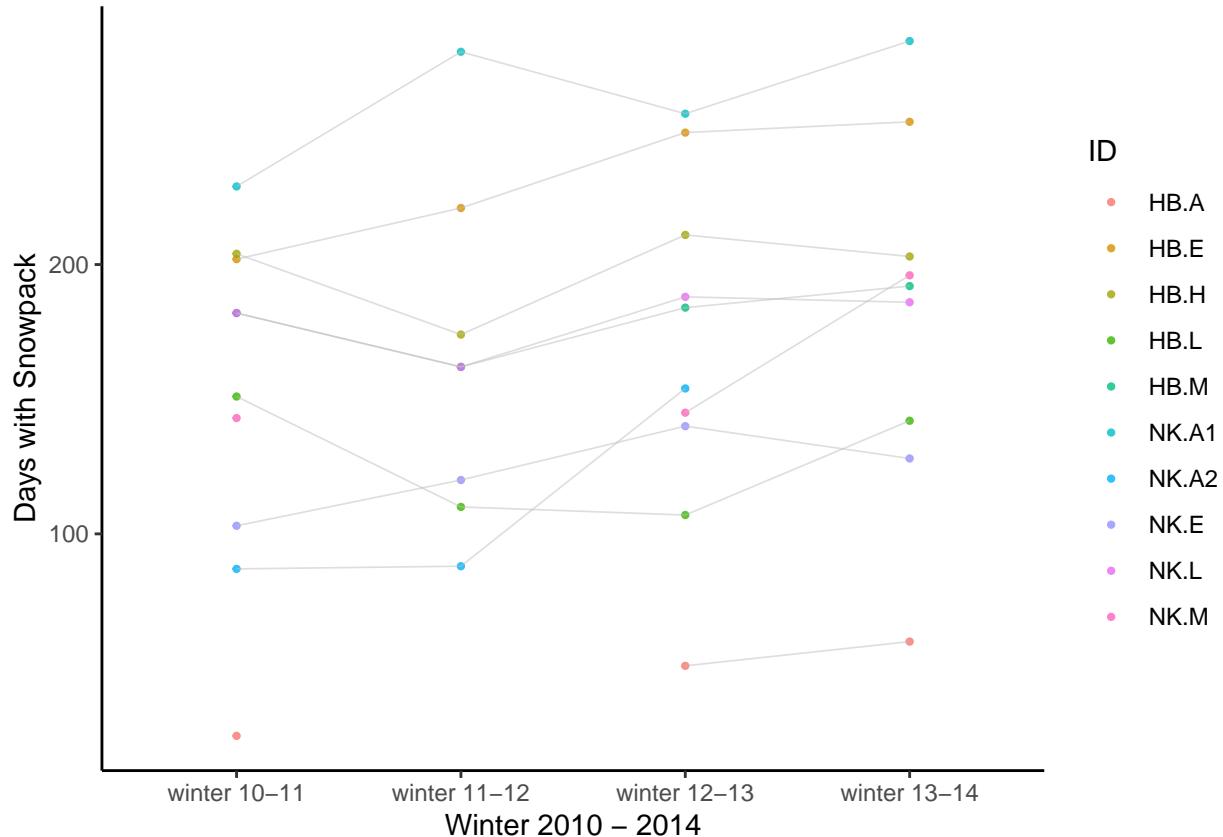
#select all rows that contains winter season info and columns related to winter
clim.win <- subset(clim.ca, subset = year %in% c("winter 10-11", "winter 11-12", "winter 12-13", "winter 13-14"))
clim.win <- subset(clim.ca, select=c(ID, year, d.snowpack, Dec.FebTav, wTmin))
head(clim.win)

##      ID      year d.snowpack Dec.FebTav wTmin
## 1  HB.A winter 10-11      25     -11.6 -22.3
## 2  HB.A winter 11-12      NA       NA     NA
## 3  HB.A winter 12-13      51     -11.3 -17.7
## 4  HB.A winter 13-14      60     -10.0 -17.7
## 5  HB.E winter 10-11     202      -1.7  -8.5
## 6  HB.E winter 11-12     221      -2.0 -10.8

ggplot(clim.win, aes(x = year, y = d.snowpack, color = ID)) +
  geom_point(size = 0.8, alpha = 0.8) +
  geom_line(aes(group = ID), color = "grey", size = 0.3, alpha = 0.5) +
  xlab("Winter 2010 - 2014") +
  ylab("Days with Snowpack") +
  theme_classic()

## Warning: Removed 3 rows containing missing values (geom_point).
## Warning: Removed 1 row(s) containing missing values (geom_path).

```



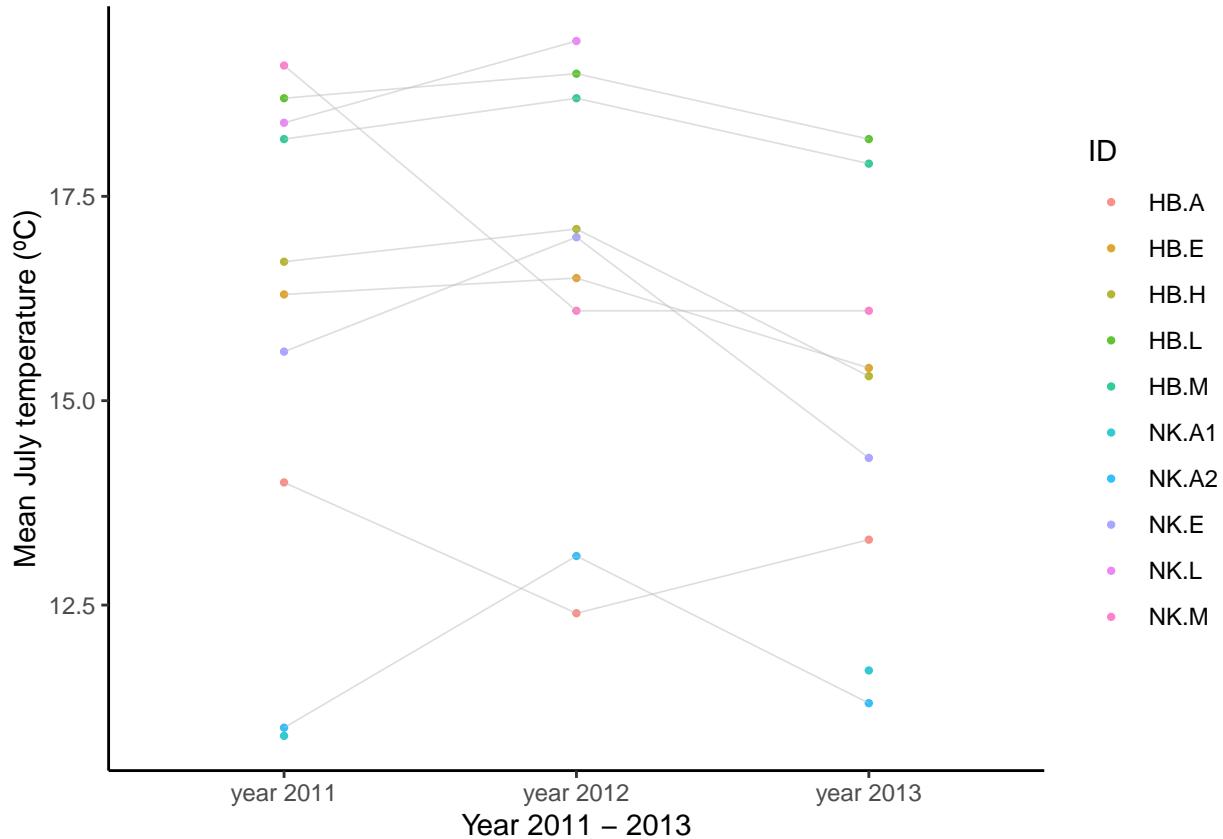
Comparison of the total days with snowpack in winter among sites sampled from elevational transect NK and HB. Grey solid lines indicate the change in days across years.

```
#select all rows that contains season info except for winter and columns related
clim.not.win <- subset(clim.ca, subset = year %in% c("year 2011", "year 2012", "year 2013"),
select=-c(d.snowpack, Dec.FebTav, wTmin))
head(clim.not.win)

##      ID      year avmaxminTJuly avTJuly d.grows GDD.grows GDD.yr
## 5  HB.A  year 2011        14.0    10.7     91       612      697
## 6  HB.A  year 2012        12.4    11.4     NA       378      461
## 7  HB.A  year 2013        13.3    12.2    111       491      532
## 12 HB.E  year 2011        16.3    13.1    105       671      671
## 13 HB.E  year 2012        16.5    13.9    102       485      660
## 14 HB.E  year 2013        15.4    13.5    110       550      576

ggplot(clim.not.win, aes(x = year, y = avmaxminTJuly, color = ID)) +
  geom_point(size = 0.8, alpha = 0.8) +
  geom_line(aes(group = ID), color = "grey", size = 0.3, alpha = 0.5) +
  xlab("Year 2011 - 2013") +
  ylab("Mean July temperature (°C)") +
  theme_classic()

## Warning: Removed 2 rows containing missing values (geom_point).
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



Comparison of the mean July temperature (°C) among sites sampled from elevational transect NK and HB. Grey solid lines indicate the change across years from 2011 to 2013.

## Visualize Climate Similarity/Disimilarity

In winter starting from 2010 to 2014

```

clim.win$tr.si.yr <- paste(clim.win$ID, clim.win$year)
row.names(clim.win) <- clim.win$tr.si.yr
clim.win <- subset(clim.win, select = -c(ID, year) )
head(clim.win)

##                                     d.snowpack Dec.FebTav wTmin          tr.si.yr
## HB.A winter 10-11           25      -11.6 -22.3 HB.A winter 10-11
## HB.A winter 11-12           NA        NA    NA HB.A winter 11-12
## HB.A winter 12-13           51      -11.3 -17.7 HB.A winter 12-13
## HB.A winter 13-14           60      -10.0 -17.7 HB.A winter 13-14
## HB.E winter 10-11          202      -1.7  -8.5 HB.E winter 10-11
## HB.E winter 11-12          221      -2.0 -10.8 HB.E winter 11-12

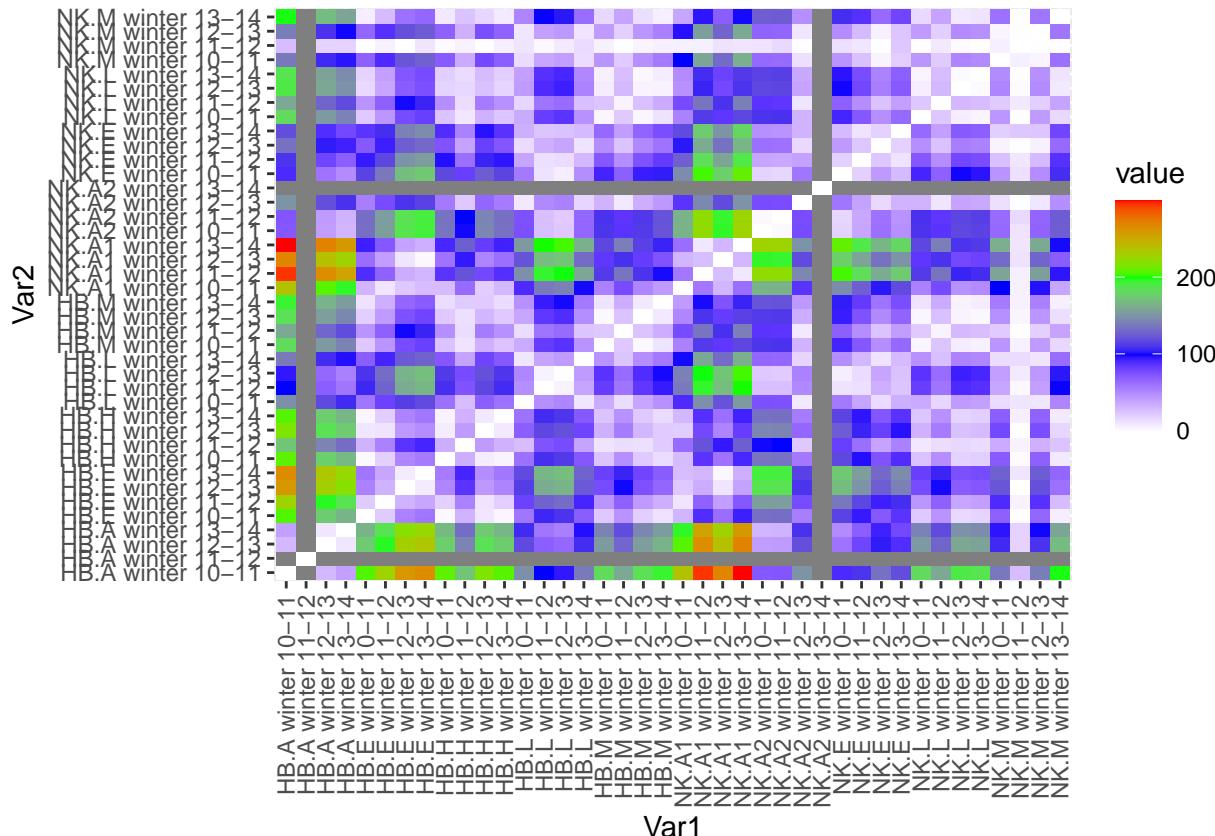
climwDist <- dist(clim.win, method = 'euclidean')

## Warning in dist(clim.win, method = "euclidean"): NAs introduced by coercion
climwDistMat <- as.matrix(climwDist)
# rearrange the data from an n×n matrix to a n2×3 matrix
climwPData <- melt(climwDistMat)

# plot the matrix

```

```
ggplot(data = climwPData, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradientn(colours = c("white", "blue", "green", "red")) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



A tile plot that demonstrates similarity/dissimilarity of climate characteristics in winter across sample locations at elevation transect NK and HB from 2010 to 2014, color gradient from purple to red indicates value from 0 to >200 and grey indicates missing data. According to the distance matrix, climates in winter at high-elevation (~100m below range edge) and high-elevation range edge at HB are mostly distinct from the rest.

### In other seasons except winter from 2011 to 2013

```
clim.not.win$tr.si.yr <- paste(clim.not.win$ID, clim.not.win$year)
row.names(clim.not.win) <- clim.not.win$tr.si.yr
clim.not.win <- subset(clim.not.win, select = -c(ID, year) )
head(clim.not.win)

##          avmaxminTJuly avTJuly d.grows GDD.grows GDD.yr      tr.si.yr
## HB.A year 2011        14.0   10.7     91       612     697 HB.A year 2011
## HB.A year 2012        12.4   11.4     NA       378     461 HB.A year 2012
## HB.A year 2013        13.3   12.2    111       491     532 HB.A year 2013
## HB.E year 2011        16.3   13.1    105       671     671 HB.E year 2011
## HB.E year 2012        16.5   13.9    102       485     660 HB.E year 2012
## HB.E year 2013        15.4   13.5    110       550     576 HB.E year 2013

climnwDist <- dist(clim.not.win, method = 'euclidean')

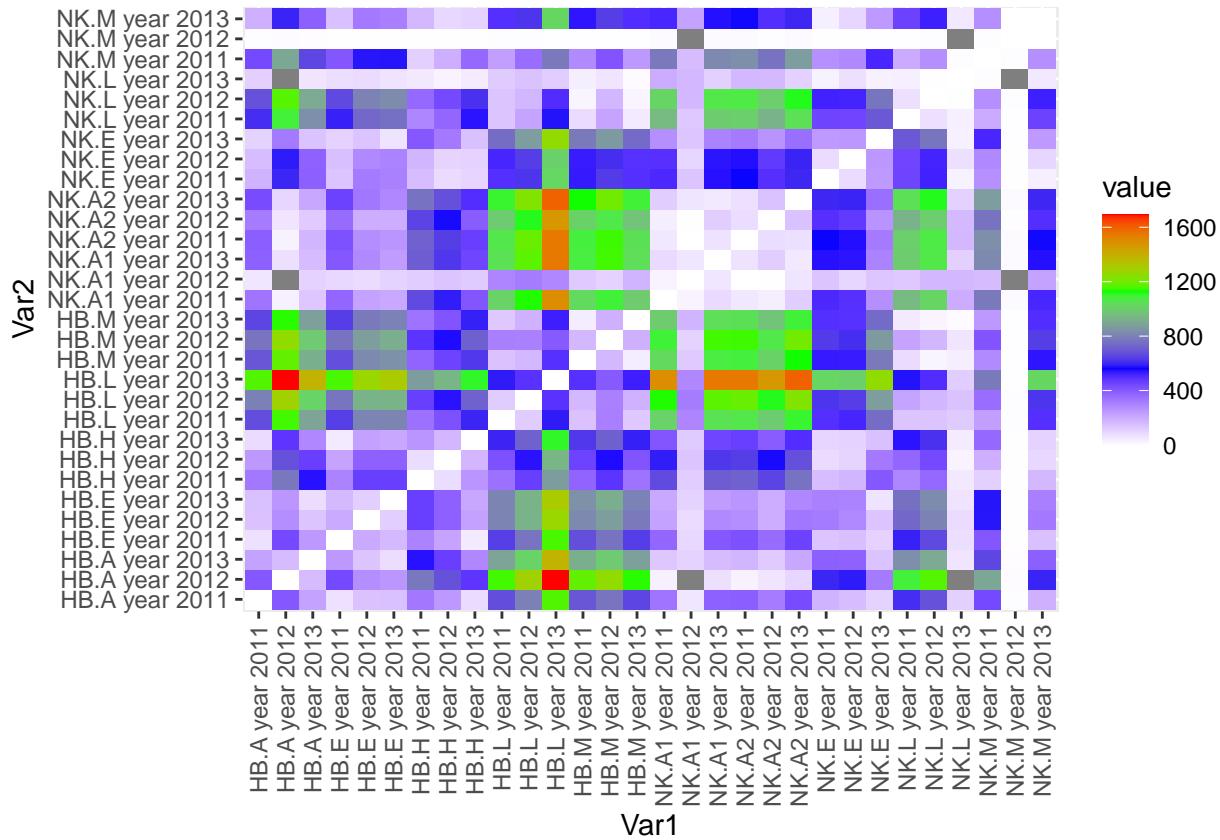
## Warning in dist(clim.not.win, method = "euclidean"): NAs introduced by coercion
```

```

climnwDistMat <- as.matrix(climnwDist)
# rearrange the data from an n×n matrix to a n2×3 matrix
climnwPData <- melt(climnwDistMat)

# plot the matrix
ggplot(data = climnwPData, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradientn(colours = c("white", "blue", "green", "red")) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))

```



A tile plot that demonstrates similarity/dissimilarity of climate characteristics in other seasons except winter across sample locations at elevation transect NK and HB from 2011 to 2013, color gradient from purple to red indicates value from 0 to >200 and grey indicates missing data. According to the distance matrix, climates at low- and mid-elevation at HB and above the high-elevation range edge at NK are mostly distinct from each other.

## Lifetime Reproductive Success Similiarity Analysis

### Description of the variables:

**year** = year plants were active

**transect** = elevational transect (where NK = Nakiska and HB = Hailstone Butte)

**site** = site seeds were transplanted into. L = low elevation, M = mid-elevation (range centre), H = high-elevation (~100 m below range edge), E or HE = high-elevation range edge, A = above range (HB), A1 = 1st site above the range at NK, A2 = second site above the range at NK

**treat** = treatment in warming experiment. TR = normal transplant (control), OTC = Open topped warming chambers (warmed), dormant = plants found > 1 year after seeds planted

**Source** = source population: low-elevation from that transect (L), mid-elevation from that transect (M), high-elevation (within 100 masl of absolute high-elevation range edge) from that transect (H), edge (absolute high-elevation range edge E), high-elevation (2000 m) populations on neighbouring Moose Mountain (MT) and Fortress Mountain (F), and seeds produced by plants transplanted above the range the previous year (A)

**plant** = plantID within each 5x5 subplot (row.column from 1.1 = topleft plant to 5.5 = bottom right plant)

**tbo** = top bitten off by mammal. 0 = no, plant intact, 1 = top bitten off the primary stem, 2 = top bitten off some secondary stalks but primary stem intact

**Ln.tot** = total leaf nodes (each leaf node produces 2 leaves), including primary, secondary and tertiary stems

**Fr.tot** = total fruits from all stems

**aveseed1** = average number of seeds per fruit from primary stem fruits

**aveseed2** = average number of seeds per fruit from secondary stem fruits

**totgoodseed** = total viable seed produced by the plant

**totbadseed** = total non-viable seed produced by the plant

## Tidy Data

```
# Select variables
ltrsHB_clean <- ltrsHB %>% select(year:treat, Source:totbadseed)
ltrsNK_clean <- ltrsNK %>% select(year:treat, Source:tbo, Ln.tot:totbadseed)

# Merge datasets
ltrsdat <- rbind(ltrsHB_clean, ltrsNK_clean)

# Remove data not from High, Medium, or Low elevation
elevation <- c("H", "M", "L")
ltrsdat <- ltrsdat %>%
  filter(Source %in% elevation)

# Examine variable classes
str(ltrsdat)

## 'data.frame': 1562 obs. of 13 variables:
## $ year      : int  2011 2011 2011 2011 2011 2011 2011 2011 2011 2011 ...
## $ transect   : chr  "HB" "HB" "HB" "HB" ...
## $ site       : chr  "A" "H" "H" "H" ...
## $ treat      : chr  "TR" "TR" "TR" "TR" ...
## $ Source     : chr  "H" "L" "M" "H" ...
## $ plant      : chr  "2.3" "2.4" "1.5" "4.2" ...
## $ tbo        : int  0 0 0 0 0 0 0 0 0 ...
## $ Ln.tot    : int  6 12 11 8 6 9 10 12 8 8 ...
## $ Fr.tot    : int  2 2 1 2 2 1 1 3 3 3 ...
## $ aveseed1  : num  4 3.75 2 4.5 6.5 ...
## $ aveseed2  : num  NA NA NA NA NA NA NA NA NA ...
## $ totgoodseed: int  8 8 2 9 13 5 3 5 22 20 ...
## $ totbadseed : num  7 1.5 0 0 0 0 0 0 0 0 ...

# Subset into High, Medium, and Low Source Elevation
ltrsHigh <- ltrsdat %>%
```

```

filter(Source == "H") %>%
group_by(transect, site, plant) %>%
mutate(dupID = row_number()) %>%
ungroup() %>%
mutate(plantID = paste(transect, site, plant, dupID), sep = "_") %>%
select(plantID, tbo:totbadseed)

ltrsMed <- ltrsdat %>%
filter(Source == "M") %>%
group_by(transect, site, plant) %>%
mutate(dupID = row_number()) %>%
ungroup() %>%
mutate(plantID = paste(transect, site, plant, dupID), sep = "_") %>%
select(plantID, tbo:totbadseed)

ltrsLow <- ltrsdat %>%
filter(Source == "L") %>%
group_by(transect, site, plant) %>%
mutate(dupID = row_number()) %>%
ungroup() %>%
mutate(plantID = paste(transect, site, plant, dupID), sep = "_") %>%
select(plantID, tbo:totbadseed)

# Replace column numbers with plantID
ltrsH_OTU <- data.frame(ltrsHigh[,-1], row.names = ltrsHigh$plantID)
ltrsM_OTU <- data.frame(ltrsMed[,-1], row.names = ltrsMed$plantID)
ltrsL_OTU <- data.frame(ltrsLow[,-1], row.names = ltrsLow$plantID)

```

## Examine Dissimilarities for High Elevation Sourced Plants

### Create Distance Matrix

```

# Create distance values
ltrsH_Dist <- dist(ltrsH_OTU, method = 'euclidean')

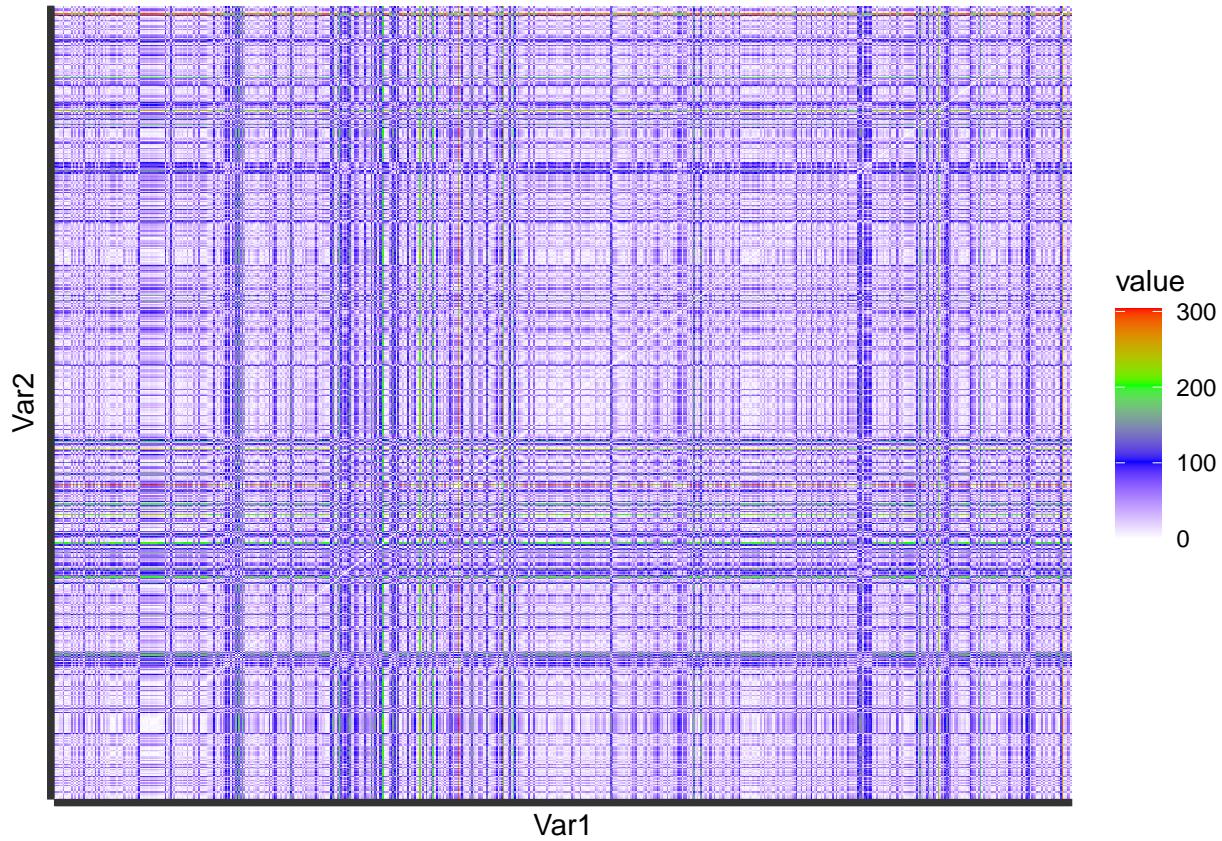
# Reshape as matrix
ltrsH_DM <- as.matrix(ltrsH_Dist)

# Reshape to be plotted
ltrsH_plotdat <- melt(ltrsH_DM)

# Distance matrix plot
HDM_Plot <- ggplot(ltrsH_plotdat, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradientn(colours = c("white", "blue", "green", "red")) +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank())

```

HDM\_Plot



### Visualise similarities with Neighbour-Joining tree

```

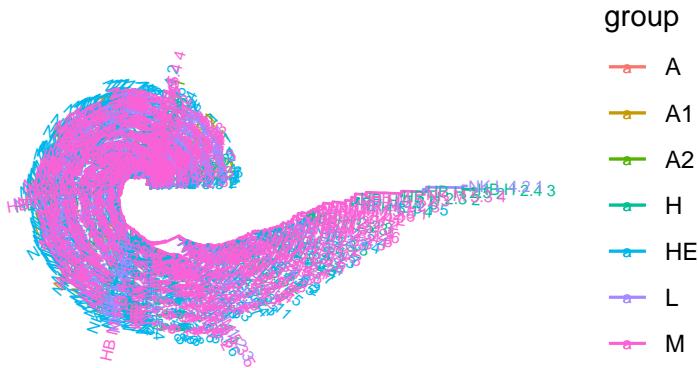
# NJ method
ltrsH_tree <- nj(ltrsH_Dist)

# Group by Site
SiteListH <- gsub("\\w+\\s+(\\w+)\\s.*", "\\1", ltrsH_tree$tip.label)
SiteGroupH <- split(ltrsH_tree$tip.label, SiteListH)
SiteTreeH <- groupOTU(ltrsH_tree, SiteGroupH)

# Create tree
ltrsHNJTree <- ggtree(SiteTreeH, layout = "circular", aes(colour = group)) +
  geom_tiplab(size = 2, aes(angle = 0)) +
  theme(legend.position = "right")

## Warning: The tree contained negative edge lengths. If you want to ignore the edges,
## you can set 'options(ignore.negative.edge=TRUE)', then re-run ggtree.
ltrsHNJTree

```



It appears that plants sourced from high elevations do not have clear similarities when transplanted to similar sites.

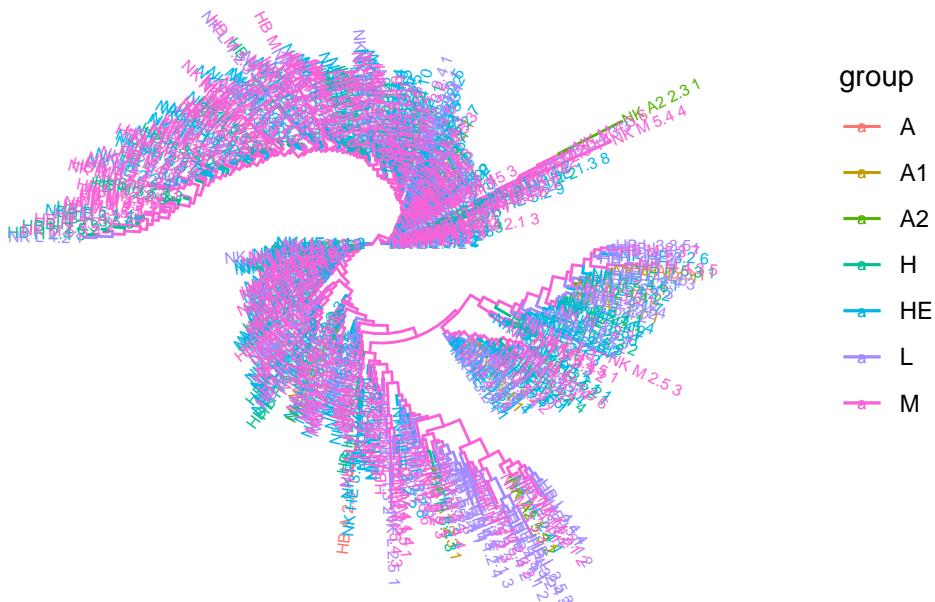
### Visualise dissimilarities with Bray-Curtis Dissimilarity

```
# Create distance matrix and tree
ltrsH_bray <- vegdist(ltrsH_OTU, method = "bray", binary = F, na.rm = T)
ltrsHTree_bray <- nj(ltrsH_bray)

# Group by Source
SiteGroupHBray <- split(ltrsHTree_bray$tip.label, SiteListH)
SiteTreeHBray <- groupOTU(ltrsHTree_bray, SiteGroupHBray)

# Create tree
ltrsHBrayTree <- ggtree(SiteTreeHBray, layout = "circular", aes(colour = group)) +
  geom_tiplab(size = 2, aes(angle = 0)) +
  theme(legend.position = "right")

## Warning: The tree contained negative edge lengths. If you want to ignore the edges,
## you can set 'options(ignore.negative.edge=TRUE)', then re-run ggtree.
ltrsHBrayTree
```



The Bray-Curtis Dissimilarity tree shows more distinct branches, and one that is notably bluer in hue, indicating that at least some high elevation sourced plants share characteristics when transplanted to high elevation or high elevation range edge sites. However, there still appears to be a through mixing of plant characters.

## Examine Dissimilarities for Medium Elevation Sourced Plants

### Create Distance Matrix

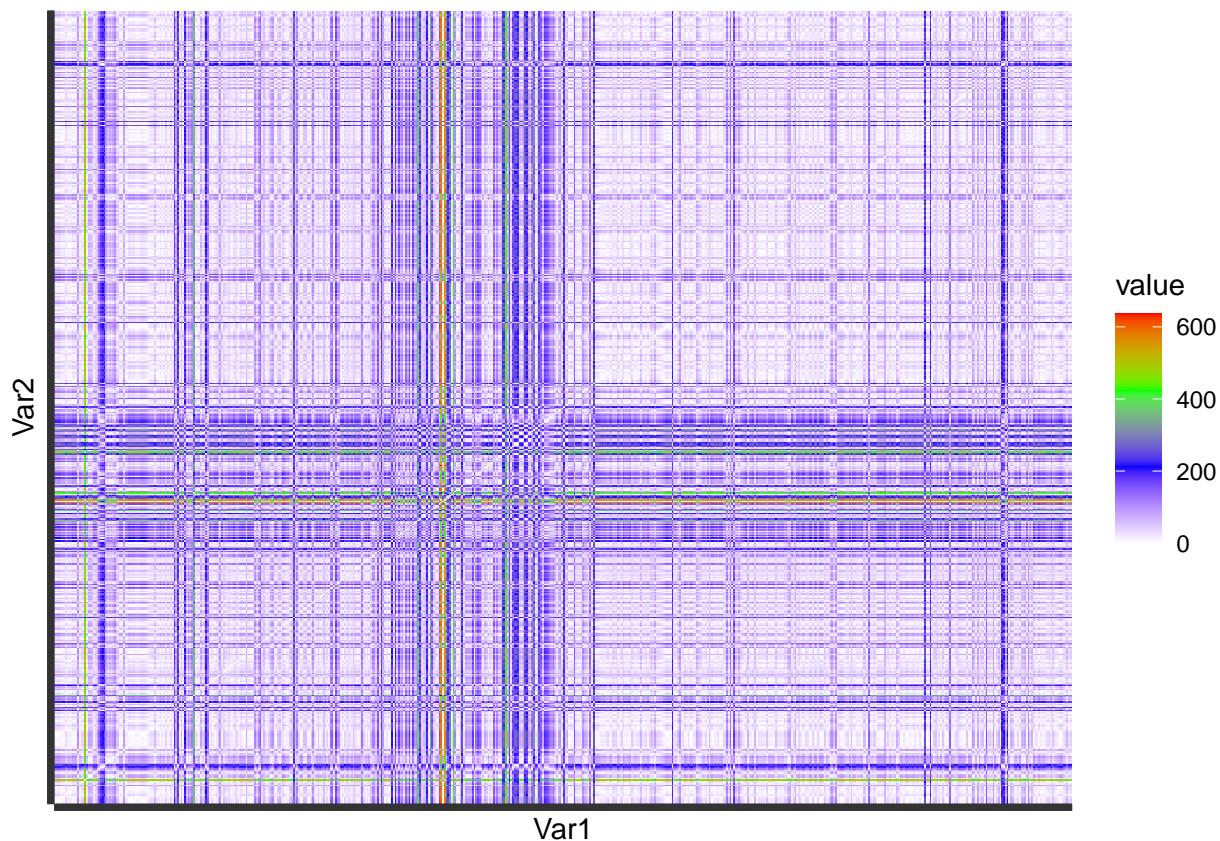
```
# Create distance values
ltrsM_Dist <- dist(ltrsM_OTU, method = 'euclidean')

# Reshape as matrix
ltrsM_DM <- as.matrix(ltrsM_Dist)

# Reshape to be plotted
ltrsM_plotdat <- melt(ltrsM_DM)

# Distance matrix plot
MDM_Plot <- ggplot(ltrsM_plotdat, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradientn(colours = c("white", "blue", "green", "red")) +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank())

MDM_Plot
```



## Visualise similarities with Neighbour-Joining tree

```

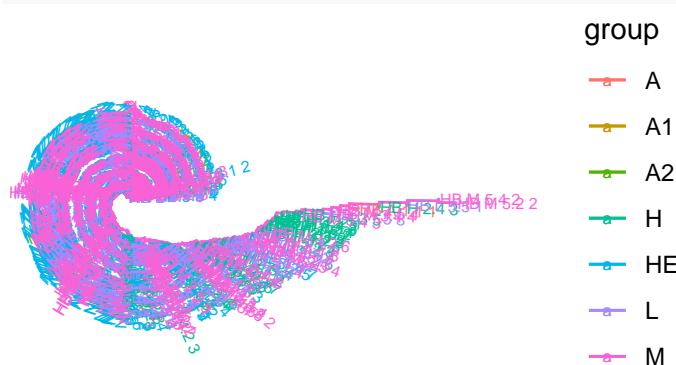
# NJ method
ltrsM_tree <- nj(ltrsM_Dist)

# Group by Source
SiteListM <- gsub("\\w+\\s+(\\w+)\\s.*", "\\1", ltrsM_tree$tip.label)
SiteGroupM <- split(ltrsM_tree$tip.label, SiteListM)
SiteTreeM <- groupOTU(ltrsM_tree, SiteGroupM)

# Create tree
ltrsMNJTree <- ggtree(SiteTreeM, layout = "circular", aes(colour = group)) +
  geom_tiplab(size = 2, aes(angle = 0)) +
  theme(legend.position = "right")

## Warning: The tree contained negative edge lengths. If you want to ignore the edges,
## you can set 'options(ignore.negative.edge=TRUE)', then re-run ggtree.

```



It appears that plants sourced from mid-elevations do not have clear similarities when transplanted to similar sites.

## Visualise dissimilarities with Bray-Curtis Dissimilarity

```

# Create distance matrix and tree
ltrsM_bray <- vegdist(ltrsM_OTU, method = "bray", binary = F, na.rm = T)
ltrsMTTree_bray <- nj(ltrsM_bray)

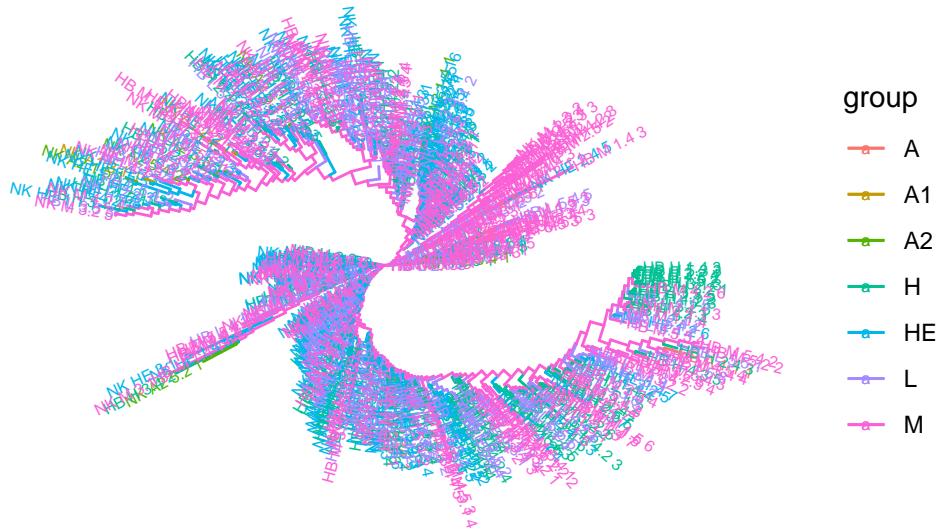
# Group by Source
SiteGroupMBray <- split(ltrsMTTree_bray$tip.label, SiteListM)
SiteTreeMBray <- groupOTU(ltrsMTTree_bray, SiteGroupMBray)

# Create tree
ltrsMBrayTree <- ggtree(SiteTreeMBray, layout = "circular", aes(colour = group)) +
  geom_tiplab(size = 2, aes(angle = 0)) +
  theme(legend.position = "right")

## Warning: The tree contained negative edge lengths. If you want to ignore the edges,
## you can set 'options(ignore.negative.edge=TRUE)', then re-run ggtree.

ltrsMBrayTree

```



There appears to be a thorough mixing of plant characters across sites, indicating that mid-elevation sourced plants transplanted to similar sites do not have similar characters.

## Examine Dissimilarities for Low Elevation Sourced Plants

## Create Distance Matrix

```

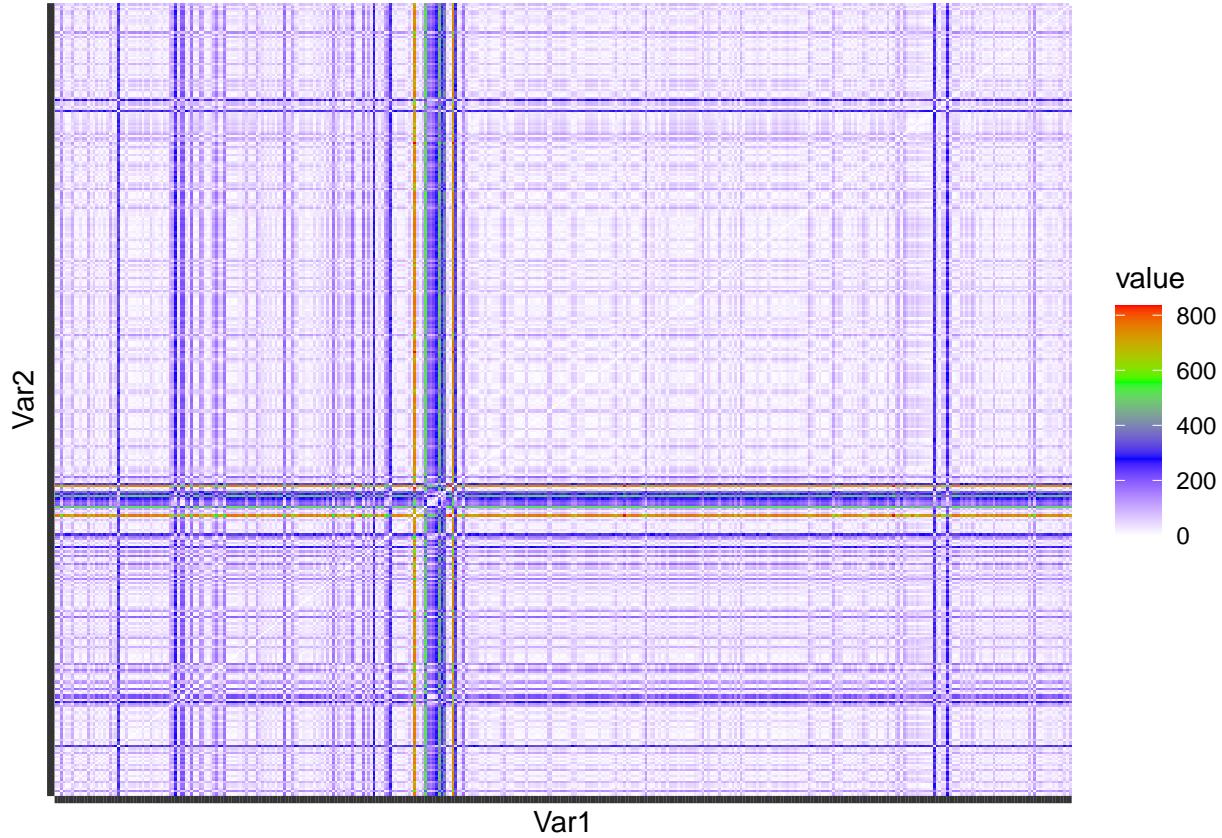
# Create distance values
ltrsL_Dist <- dist(ltrsL_OTU, method = 'euclidean')

# Reshape as matrix
ltrsL_DM <- as.matrix(ltrsL_Dist)

# Reshape to be plotted
ltrsL_plotdat <- melt(ltrsL_DM)

# Distance matrix plot
LDM_Plot <- ggplot(ltrsL_plotdat, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradientn(colours = c("white", "blue", "green", "red")) +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank())

```



### Visualise similarities with Neighbour-Joining tree

```

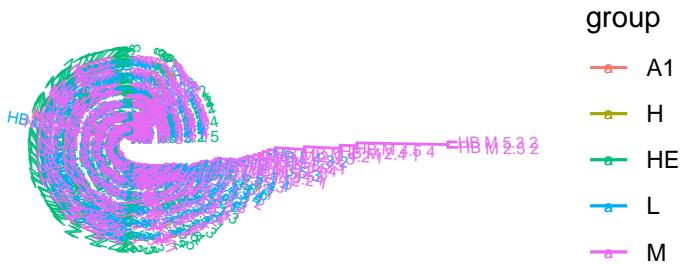
# NJ method
ltrsL_tree <- nj(ltrsL_Dist)

# Group by Source
SiteListL <- gsub("\\w+\\s+(\\w+)\\s.*", "\\1", ltrsL_tree$tip.label)
SiteGroupL <- split(ltrsL_tree$tip.label, SiteListL)
SiteTreeL <- groupOTU(ltrsL_tree, SiteGroupL)

# Create tree
ltrsLNJTree <- ggtree(SiteTreeL, layout = "circular", aes(colour = group)) +
  geom_tiplab(size = 2, aes(angle = 0)) +
  theme(legend.position = "right")

## Warning: The tree contained negative edge lengths. If you want to ignore the edges,
## you can set 'options(ignore.negative.edge=TRUE)', then re-run ggtree.
ltrsLNJTree

```



It appears that plants sourced from low elevations do not have clear similarities when transplanted to similar sites.

### Visualise dissimilarities with Bray-Curtis Dissimilarity

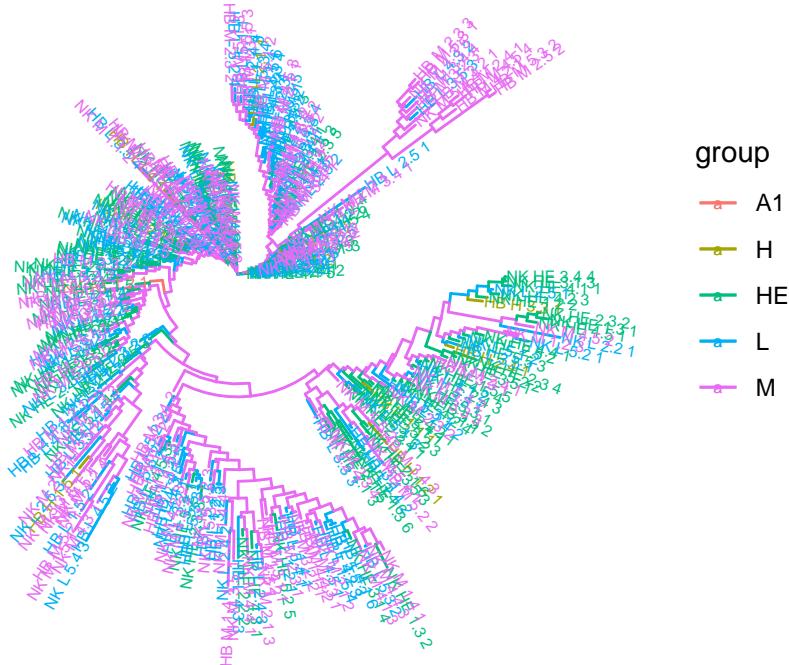
```
# Create distance matrix and tree
ltrsL_bray <- vegdist(ltrsL_OTU, method = "bray", binary = F, na.rm = T)
ltrsLTree_bray <- nj(ltrsL_bray)

# Group by Source
SiteGroupLBray <- split(ltrsLTree_bray$tip.label, SiteListL)
SiteTreeLBray <- groupOTU(ltrsLTree_bray, SiteGroupLBray)

# Create tree
ltrsLBrayTree <- ggtree(SiteTreeLBray, layout = "circular", aes(colour = group)) +
  geom_tiplab(size = 2, aes(angle = 0)) +
  theme(legend.position = "right")

## Warning: The tree contained negative edge lengths. If you want to ignore the edges,
## you can set 'options(ignore.negative.edge=TRUE)', then re-run ggtree.

ltrsLBrayTree
```



While this Bray-Curtis Dissimilarity tree does show mixing of site traits across the tree, there are branches that are more distinctly coloured. This indicates that some low elevation sourced plants may maintain similar

characteristics when transplanted to similar sites.

## Compare lifetime reproductive success between treatments across sites

### Compare leaf nodes between treatments

```
# Create plot data
ltrsPlotDat <- ltrsdat %>% mutate(propgoodseed = totgoodseed/(totbadseed + totgoodseed)) %>%
  select(Source, site, treat, Ln.tot, Fr.tot, propgoodseed)

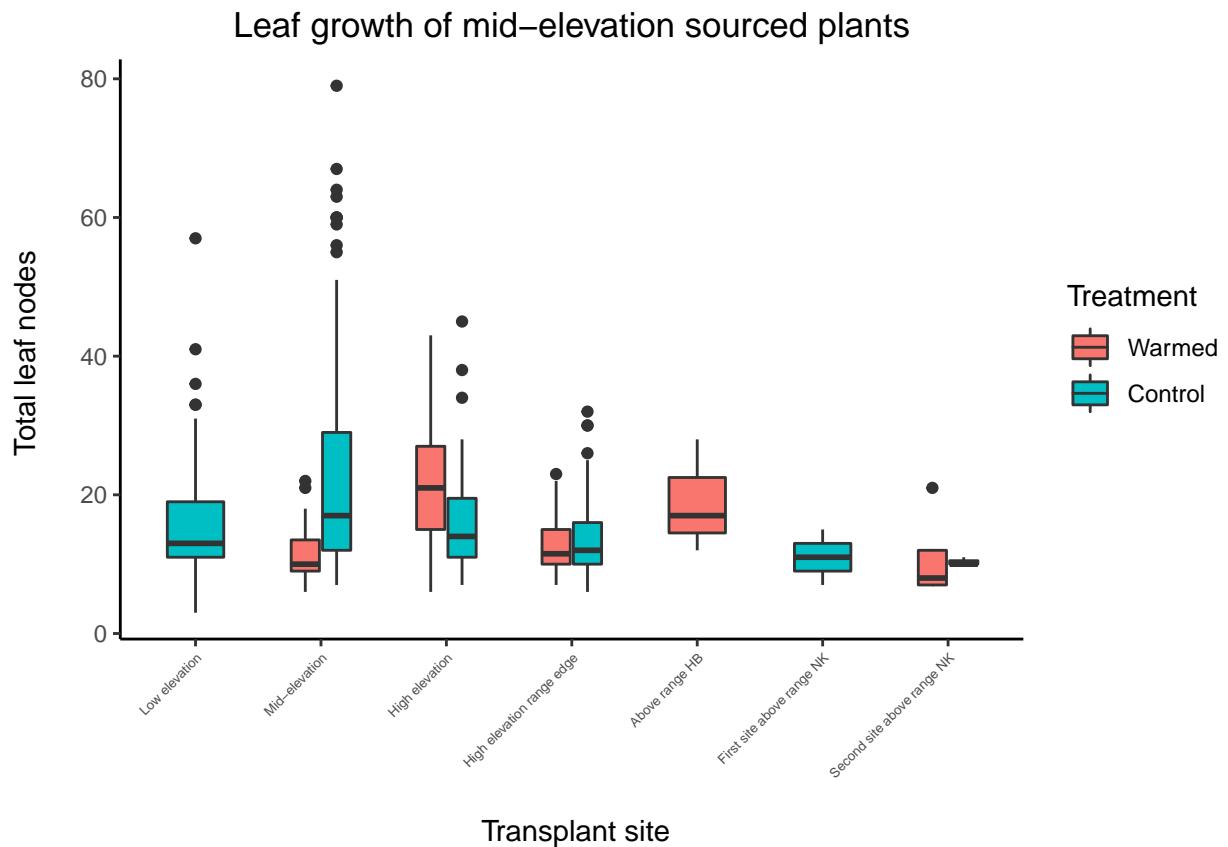
# Reorder sites
ltrsPlotDat$site <- factor(ltrsPlotDat$site , levels = c("L", "M", "H", "HE", "A", "A1", "A2"))

# Filter for mid and high elevations
MidPlotDat <- ltrsPlotDat %>% filter(Source == "M")
HighPlotDat <- ltrsPlotDat %>% filter(Source == "H")

# Mid Elevation Source Plot
# Create Mid Elevation plot
MidLeafPlot <- ggplot(MidPlotDat, aes(x = site, y = Ln.tot, fill = treat)) +
  geom_boxplot(width = 0.5) +
  scale_x_discrete(name = "\n Transplant site",
                    labels = c("Low elevation", "Mid-elevation", "High elevation",
                               "High elevation range edge", "Above range HB",
                               "First site above range NK", "Second site above range NK")) +
  scale_y_continuous(name = "Total leaf nodes \n")+
  scale_fill_discrete(name = "Treatment", labels = c("Warmed", "Control")) +
  ggtitle("Leaf growth of mid-elevation sourced plants") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 5),
        plot.title = element_text(hjust = 0.5))

MidLeafPlot

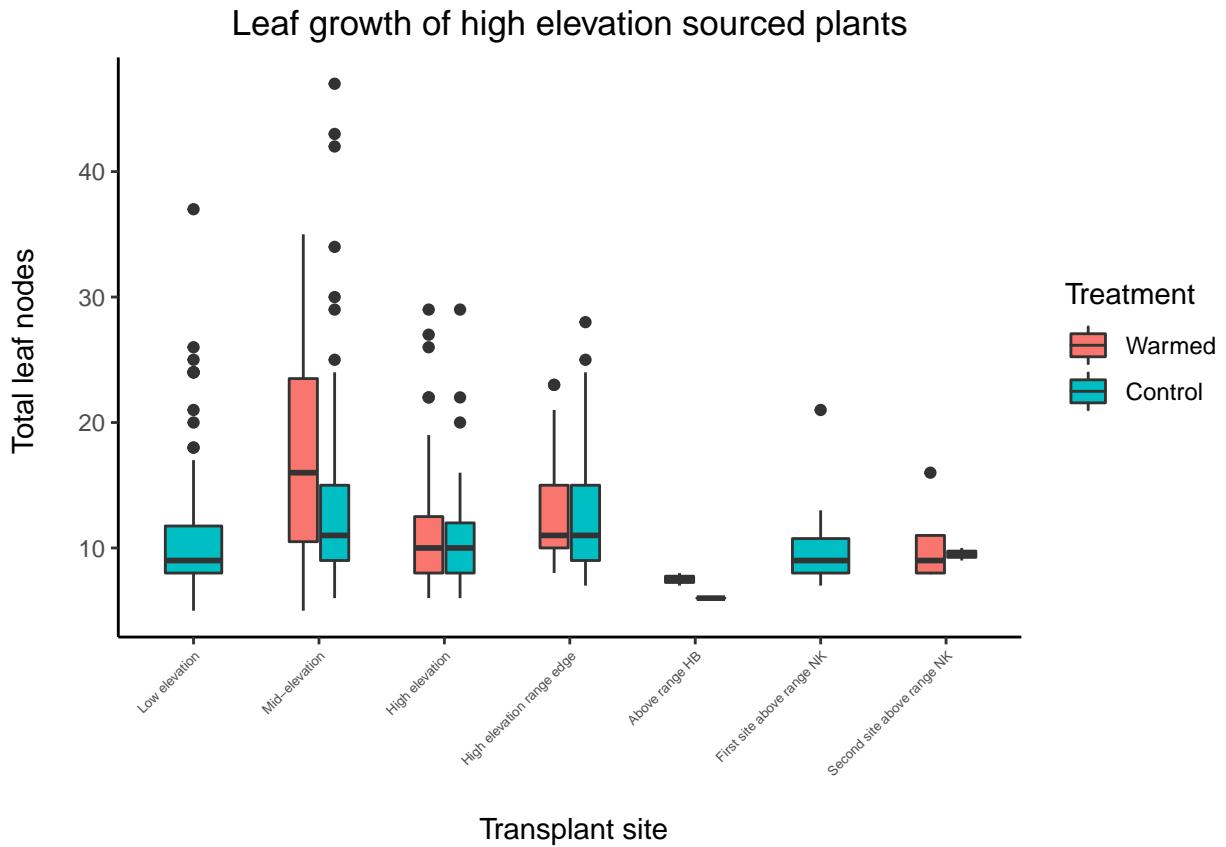
## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
```



```
# High Elevation Source Plot
# Create Mid Elevation plot
HighLeafPlot <- ggplot(HighPlotDat, aes(x = site, y = Ln.tot, fill = treat)) +
  geom_boxplot(width = 0.5) +
  scale_x_discrete(name = "\n Transplant site",
                    labels = c("Low elevation", "Mid-elevation", "High elevation",
                              "High elevation range edge", "Above range HB",
                              "First site above range NK", "Second site above range NK")) +
  scale_y_continuous(name = "Total leaf nodes \n")+
  scale_fill_discrete(name = "Treatment",
                      labels = c("Warmed", "Control")) +
  ggttitle("Leaf growth of high elevation sourced plants") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 5),
        plot.title = element_text(hjust = 0.5))

HighLeafPlot

## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```

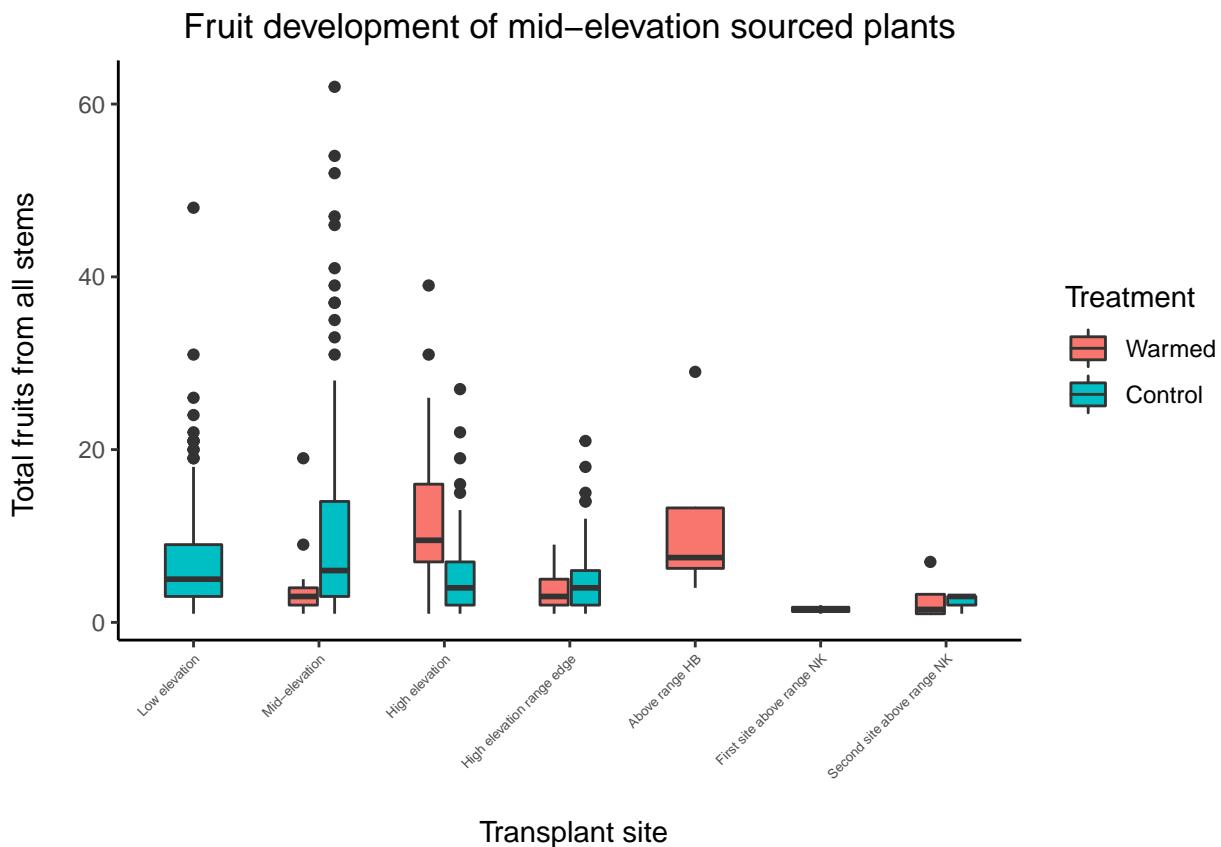


The boxplot showing leaf growth data for mid-elevation plants appears to show that some differences do exist for plants transplanted to mid-elevation and high elevation sites. Mid-elevation to mid-elevation plants appear to have decreased leaf growth in warming chambers compared to the control, indicating that the conditions for growth are not within the species optimal. Conversely, mid-elevation plants transplanted to high elevation with warming showed higher levels of leaf growth than the control, likely due to the warming chamber producing a temperature closer to their native range. The boxplot showing leaf growth data for high elevation plants appears to show that there may have been a difference in leaf growth for high elevation sourced plants transplanted to mid-elevation sites. When placed in a warming chamber, these plants appear to have slightly elevated levels of leaf growth compared to the control, however there is a high degree of variability.

### Compare fruits between treatments

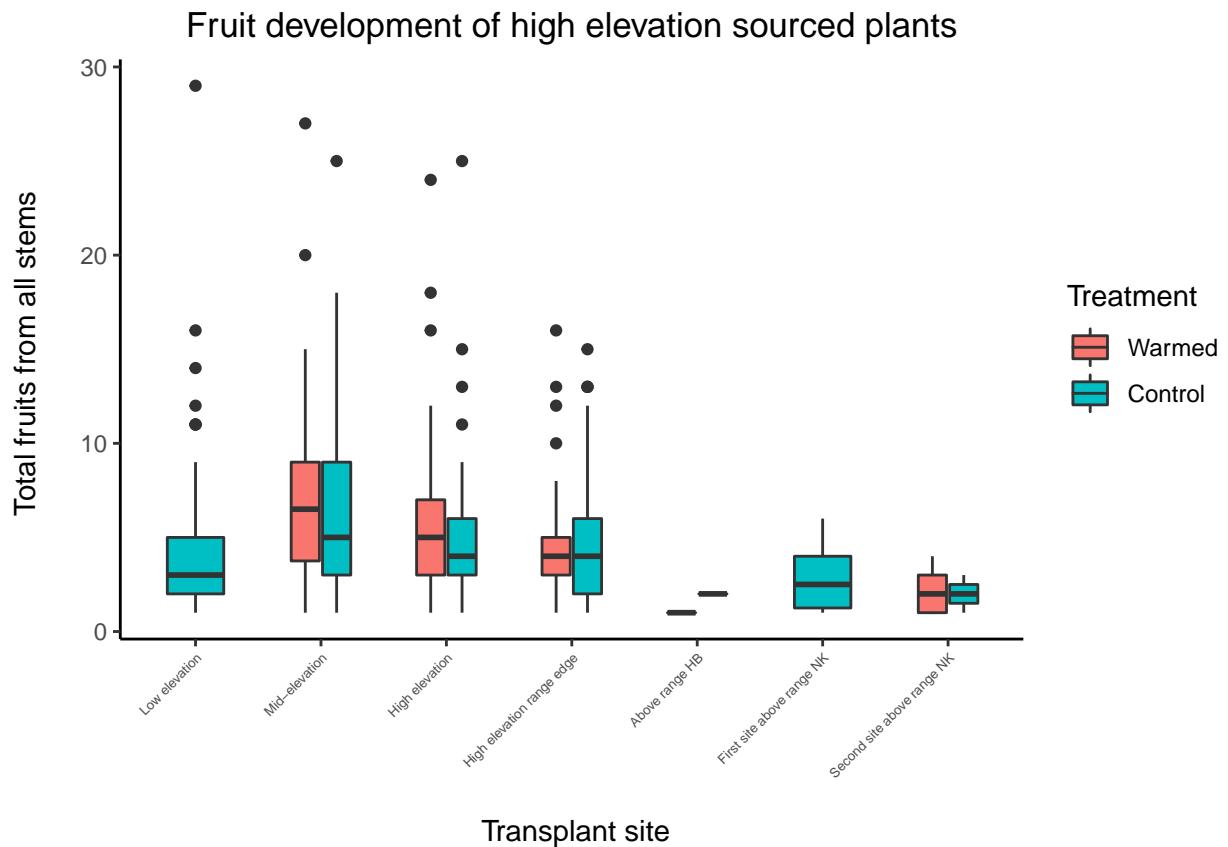
```
# Mid Elevation Source Plot
# Create Mid Elevation plot
MidFruitPlot <- ggplot(MidPlotData, aes(x = site, y = Fr.tot, fill = treat)) +
  geom_boxplot(width = 0.5) +
  scale_x_discrete(name = "\n Transplant site",
                    labels = c("Low elevation", "Mid-elevation", "High elevation",
                              "High elevation range edge", "Above range HB",
                              "First site above range NK", "Second site above range NK")) +
  scale_y_continuous(name = "Total fruits from all stems \n")+
  scale_fill_discrete(name = "Treatment", labels = c("Warmed", "Control")) +
  ggtitle("Fruit development of mid-elevation sourced plants") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 5),
        plot.title = element_text(hjust = 0.5))
```

MidFruitPlot



```
# High Elevation Source Plot
# Create Mid Elevation plot
HighFruitPlot <- ggplot(HighPlotDat, aes(x = site, y = Fr.tot, fill = treat)) +
  geom_boxplot(width = 0.5) +
  scale_x_discrete(name = "\n Transplant site",
                    labels = c("Low elevation", "Mid-elevation", "High elevation",
                              "High elevation range edge", "Above range HB",
                              "First site above range NK", "Second site above range NK")) +
  scale_y_continuous(name = "Total fruits from all stems \n")+
  scale_fill_discrete(name = "Treatment",
                      labels = c("Warmed", "Control")) +
  ggtitle("Fruit development of high elevation sourced plants") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 5),
        plot.title = element_text(hjust = 0.5))
```

HighFruitPlot

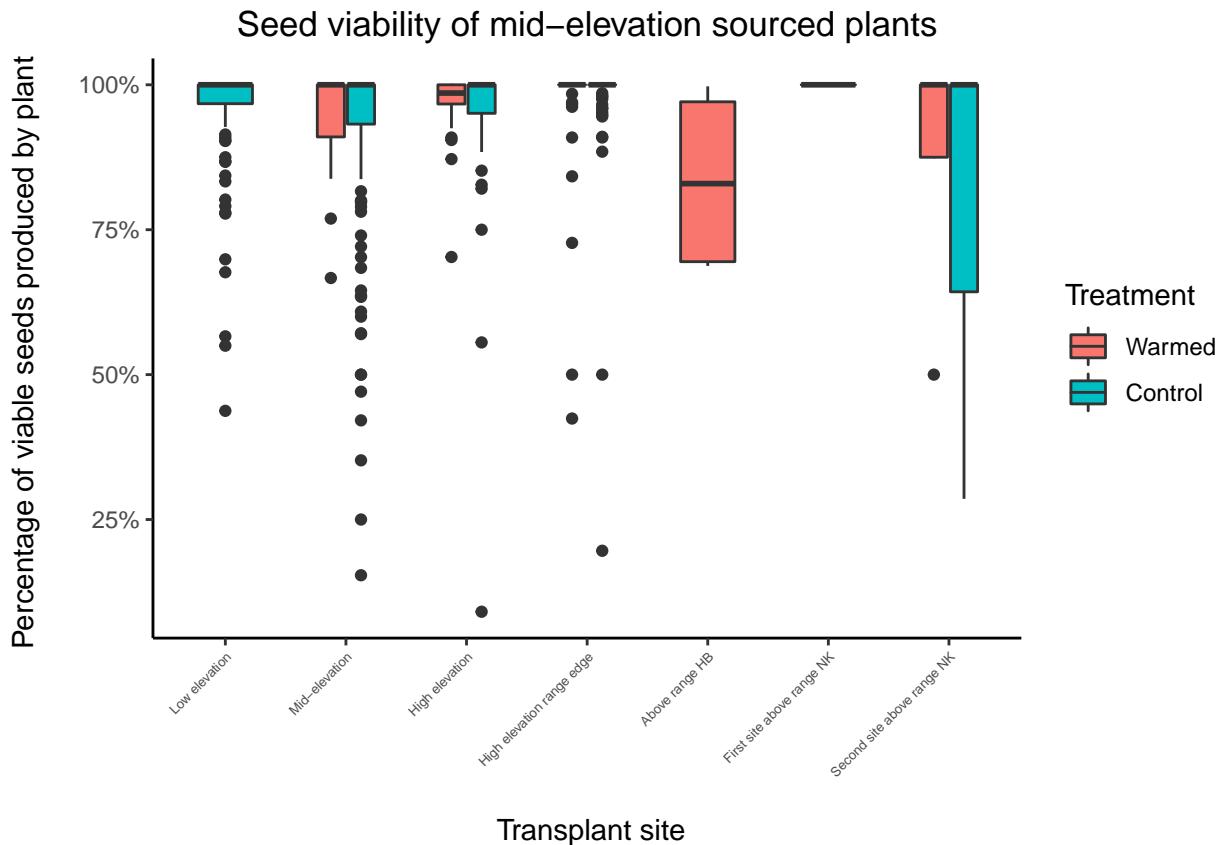


Similar to the pattern observed in leaf growth of mid-elevation plants transplanted to mid- and high elevation sites, mid-elevation plants transplanted to mid- and high elevation sites show possible differences in fruit development. Possibly as a result of growing in elevated temperatures outside of the plant's native range, plants transplanted to mid-elevation warming chambers appear to have decreased fruit growth compared to controls. Plants transplanted to high elevation appear to have a much higher amount of fruits on average, possibly due to being grown in a more optimal temperature range. The boxplot showing fruit development for high elevation sourced plants appears to not illustrate many differences between sites or between treated groups and controls. The largest difference appears to be in plants transplanted above the range at Hailstone Butte, but this may be the result of a lack of data at that site.

#### Compare proportion of good seeds between sites for each source

```
# Mid Elevation Source Plot
# Create Mid Elevation plot
MidSeedPlot <- ggplot(MidPlotDat, aes(x = site, y = propgoodseed, fill = treat)) +
  geom_boxplot(width = 0.5) +
  scale_x_discrete(name = "\n Transplant site",
    labels = c("Low elevation", "Mid-elevation", "High elevation",
              "High elevation range edge", "Above range HB",
              "First site above range NK", "Second site above range NK")) +
  scale_y_continuous(labels = scales::percent,
    name = "Percentage of viable seeds produced by plant \n")+
  scale_fill_discrete(name = "Treatment", labels = c("Warmed", "Control")) +
  ggtitle("Seed viability of mid-elevation sourced plants") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 5),
    plot.title = element_text(hjust = 0.5))
```

MidSeedPlot

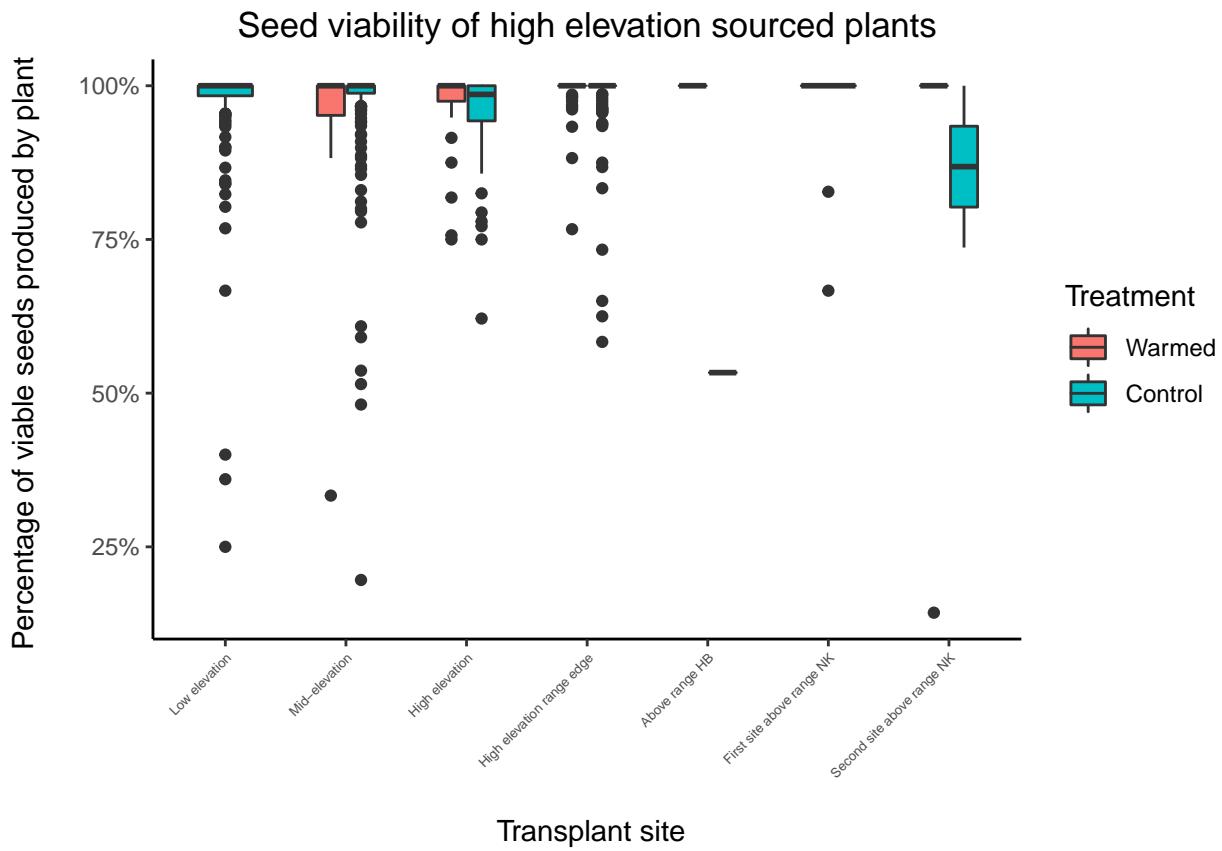


```
# High Elevation Source Plot
```

```
# Create Mid Elevation plot
```

```
HighSeedPlot <- ggplot(HighPlotDat, aes(x = site, y = propgoodseed, fill = treat)) +  
  geom_boxplot(width = 0.5) +  
  scale_x_discrete(name = "\n Transplant site",  
    labels = c("Low elevation", "Mid-elevation", "High elevation",  
    "High elevation range edge", "Above range HB",  
    "First site above range NK", "Second site above range NK")) +  
  scale_y_continuous(labels = scales::percent,  
    name = "Percentage of viable seeds produced by plant \n") +  
  scale_fill_discrete(name = "Treatment",  
    labels = c("Warmed", "Control")) +  
  ggtitle("Seed viability of high elevation sourced plants") +  
  theme_classic() +  
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 5),  
    plot.title = element_text(hjust = 0.5))
```

HighSeedPlot



In both mid-elevation and high elevation sourced plants, there appears to be little differences in treatments in proportion viable seeds produced. There is, however, large amounts of variation for low to high elevation range edge sites, and a notable decrease in amount of data available for above range sites at both Nakiska and Hailstone Butte.

## PCA

Using ‘totgoodseed’ to measure the indirect fitness of the sampling plant communities.

```
head(ltrsdat)
```

```
##   year transect site treat Source plant tbo Ln.tot Fr.tot aveseed1 aveseed2
## 1 2011      HB    A   TR     H   2.3   0     6     2    4.00    NA
## 2 2011      HB    H   TR     L   2.4   0    12     2    3.75    NA
## 3 2011      HB    H   TR     M   1.5   0    11     1    2.00    NA
## 4 2011      HB    H   TR     H   4.2   0     8     2    4.50    NA
## 5 2011      HB    H   TR     H   5.2   0     6     2    6.50    NA
## 6 2011      HB    H   TR     M   2.4   0     9     1    5.00    NA
##   totgoodseed totbadseed
## 1          8       7.0
## 2          8       1.5
## 3          2       0.0
## 4          9       0.0
## 5         13       0.0
## 6          5       0.0
```

```
ltrs_pc <- subset(ltrsdat, subset = site %in% c("L", "M", "H"),
                    select=-c(aveseed2, totbadseed))
```

```

head(ltrs_pc)

##   year transect site treat Source plant tbo Ln.tot Fr.tot aveseed1 totgoodseed
## 2 2011      HB    H    TR     L   2.4   0    12     2    3.75        8
## 3 2011      HB    H    TR     M   1.5   0    11     1    2.00        2
## 4 2011      HB    H    TR     H   4.2   0     8     2    4.50        9
## 5 2011      HB    H    TR     H   5.2   0     6     2    6.50       13
## 6 2011      HB    H    TR     M   2.4   0     9     1    5.00        5
## 7 2011      HB    H    TR     M   4.1   0    10     1    3.00        3

```

performs a principal components analysis on the numeric data matrix

The following chunks are used to make table 1 attached in the poster.

```

# delete NA values
ltrs_pc <- ltrs_pc %>%
  na.omit()

ltrsfull <- princomp(ltrs_pc[,8:10], cor = T)
summary(ltrsfull)

## Importance of components:
##                               Comp.1    Comp.2    Comp.3
## Standard deviation      1.3773528 0.9896383 0.35144747
## Proportion of Variance 0.6323669 0.3264613 0.04117177
## Cumulative Proportion  0.6323669 0.9588282 1.00000000
loadings(ltrsfull)

##
## Loadings:
##           Comp.1  Comp.2  Comp.3
## Ln.tot      0.693   0.167   0.701
## Fr.tot      0.701   -0.710
## aveseed1    0.165  -0.984
##
##           Comp.1  Comp.2  Comp.3
## SS loadings  1.000   1.000   1.000
## Proportion Var 0.333   0.333   0.333
## Cumulative Var 0.333   0.667   1.000

head(ltrsfull$scores)

##           Comp.1    Comp.2    Comp.3
## 2 -0.9597902 0.93187317 0.09629371
## 3 -1.1892531 1.39229285 0.08126676
## 4 -1.1827679 0.66257411 -0.14931529
## 5 -1.2190062 0.07863529 -0.23975004
## 6 -1.1791833 0.53186831  0.01075185
## 7 -1.2073722 1.10032344  0.03604938

round(cor(ltrsfull$scores),3)

##           Comp.1  Comp.2  Comp.3
## Comp.1      1      0      0
## Comp.2      0      1      0
## Comp.3      0      0      1

```

```

library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
eig.val1 <- get_eigenvalue(ltrsfull)
eig.val1

##      eigenvalue variance.percent cumulative.variance.percent
## Dim.1  1.8971008       63.236693             63.23669
## Dim.2  0.9793839       32.646130             95.88282
## Dim.3  0.1235153       4.117177            100.00000

PC axes are uncorrelated with each other: that the diagonal is 1 represents the correlation of each PC axis with itself, while the off-diagonals are zero, meaning that the correlation coefficient is <0.001. Besides, for the dimensions, an eigenvalue > 1 indicates that the principal component (PCs) accounts for more variance than accounted by one of the original variables in standardized data.

ltrs_pc$PC1 <- ltrsfull$scores[,1]
ltrs_pc$PC2 <- ltrsfull$scores[,2]
ltrs_pc$PC3 <- ltrsfull$scores[,3]

PCltrsMod1 <- lm(PC1 ~ site*treat, data = ltrs_pc)
summary(PCltrsMod1)

##
## Call:
## lm(formula = PC1 ~ site * treat, data = ltrs_pc)
##
## Residuals:
##      Min    1Q Median    3Q   Max 
## -1.6447 -0.8021 -0.3608  0.3030 10.2203 
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.29038   0.12938   2.244  0.02501 *  
## siteL        0.06262   0.14723   0.425  0.67067    
## siteM       -0.64068   0.26334  -2.433  0.01514 *  
## treatTR     -0.58964   0.18297  -3.223  0.00131 ** 
## siteL:treatTR NA        NA        NA        NA      
## siteM:treatTR 1.17600   0.30061   3.912  9.74e-05 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 1.357 on 1053 degrees of freedom
## Multiple R-squared:  0.03401,   Adjusted R-squared:  0.03034 
## F-statistic: 9.269 on 4 and 1053 DF,  p-value: 2.312e-07

anova(PCltrsMod1)

## Analysis of Variance Table
##
## Response: PC1
##           Df  Sum Sq Mean Sq F value    Pr(>F)    
## site        2   38.02 19.0095 10.3241 3.628e-05 ***
## treat       1    2.07  2.0713  1.1249  0.2891    
## site:treat  1   28.18 28.1784 15.3038 9.740e-05 *** 
## Residuals 1053 1938.86  1.8413

```

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

p-value <0.05 suggests there is no significant evidence in favor of a potential relationship between component 1 and factors including treatment type and elevation sites except median-elevation. Specifically, the linear regression line is steepest for the plants normally transplanted(TR) to median elevation.

```

PCltrsMod2 <- lm(PC2 ~ site*treat, data = ltrs_pc)
summary(PCltrsMod2)

```

```

##
## Call:
## lm(formula = PC2 ~ site * treat, data = ltrs_pc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44708 -0.64517 -0.04933  0.71707  2.14824
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.75335  0.08959 -8.409 < 2e-16 ***
## siteL        0.39547  0.10195  3.879 0.000111 ***
## siteM        1.33130  0.18235  7.301 5.63e-13 ***
## treatTR      0.39633  0.12670  3.128 0.001807 **
## siteL:treatTR NA        NA        NA        NA
## siteM:treatTR -0.77063  0.20816 -3.702 0.000225 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9396 on 1053 degrees of freedom
## Multiple R-squared:  0.1028, Adjusted R-squared:  0.0994
## F-statistic: 30.16 on 4 and 1053 DF, p-value: < 2.2e-16
anova(PCltrsMod2)

```

```

## Analysis of Variance Table
##
## Response: PC2
##             Df Sum Sq Mean Sq F value    Pr(>F)
## site          2  93.35  46.677 52.8692 < 2.2e-16 ***
## treat         1   1.07   1.073  1.2158 0.2704301
## site:treat   1  12.10  12.100 13.7057 0.0002249 ***
## Residuals  1053 929.66   0.883
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

p-value <0.05 suggests there is significant evidence in favor of a potential relationship between component 2 and factors including elevation and treatment.

```

PCltrsMod3 <- lm(PC3 ~ site*treat, data = ltrs_pc)
summary(PCltrsMod3)

```

```

##
## Call:
## lm(formula = PC3 ~ site * treat, data = ltrs_pc)
##
## Residuals:
##
```

```

##      Min     1Q   Median     3Q    Max
## -1.60769 -0.18467 -0.03685  0.16615  2.01888
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.07573   0.03348 -2.262  0.0239 *
## siteL       -0.01460   0.03810 -0.383  0.7017
## siteM        0.04716   0.06814  0.692  0.4891
## treatTR      0.08775   0.04734  1.854  0.0641 .
## siteL:treatTR NA        NA      NA      NA
## siteM:treatTR -0.03832  0.07778 -0.493  0.6224
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3511 on 1053 degrees of freedom
## Multiple R-squared: 0.006619, Adjusted R-squared: 0.002846
## F-statistic: 1.754 on 4 and 1053 DF, p-value: 0.1359
anova(PCltrsMod3)

## Analysis of Variance Table
##
## Response: PC3
##             Df  Sum Sq Mean Sq F value Pr(>F)
## site          2  0.362  0.18117  1.4696 0.23049
## treat         1  0.473  0.47274  3.8346 0.05047 .
## site:treat   1  0.030  0.02991  0.2427 0.62240
## Residuals 1053 129.814  0.12328
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

p-value >0.05 suggests there is no significant evidence in favor of a potential relationship between component 3 and factors including elevation and treatment.

**Table 1**

Table 1. Principle component analysis				
	Eigenvalue	Comp. 1	Comp. 2	Comp. 3
Proportion*	0.632	0.326	0.041	
<i>Loadings</i>				
Total leaf nodes	1.897	0.693	0.167	0.701
Total fruits	0.979	0.701		-0.710
Average seeds	0.123	-0.165	-0.984	

\*Proportion of variance

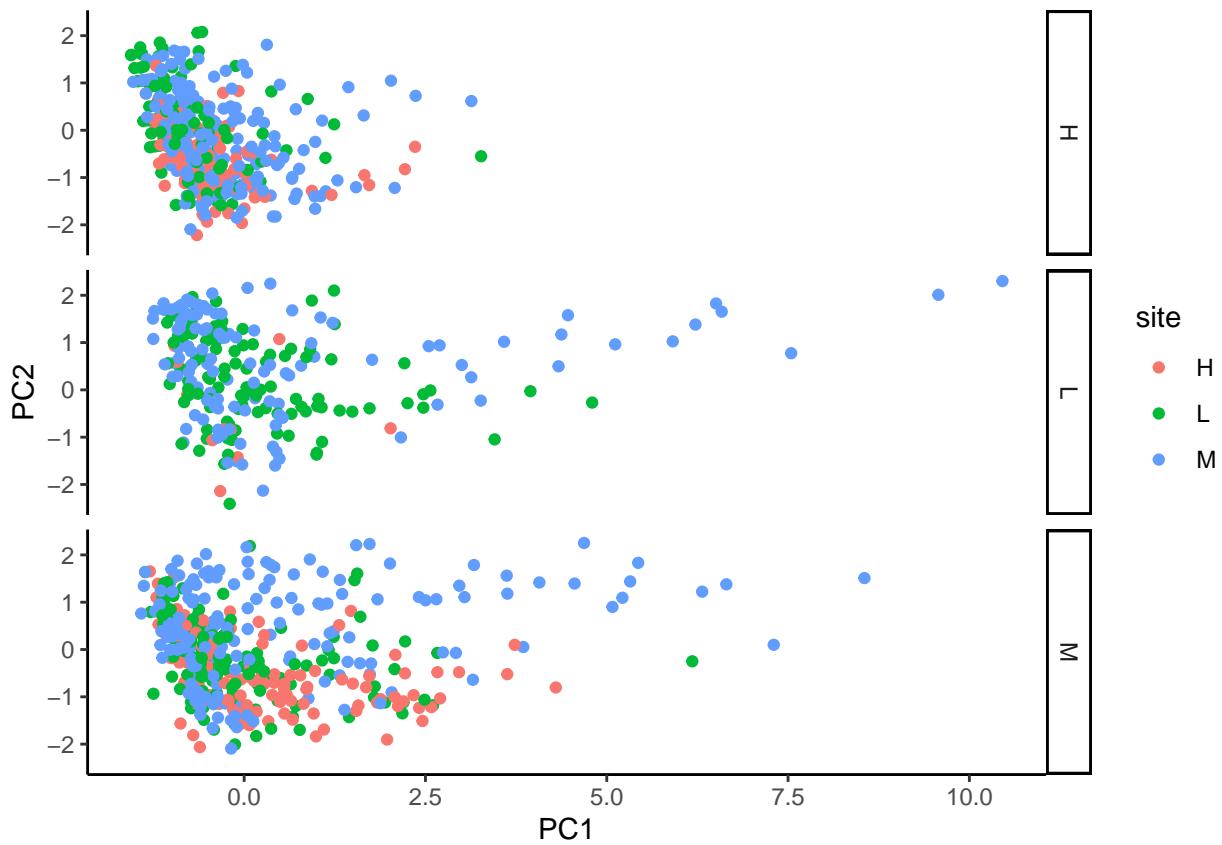
Given by the table, in the principal component analysis, the most prominent biological factor (eigenvalue: 1.897, total variance explained: 63.2%) was total leaf nodes, including primary, secondary and tertiary stems.

## Ploting PCA

```

qplot(x = PC1, y = PC2, color = site, facets = "Source", data = ltrs_pc) +
theme_classic()

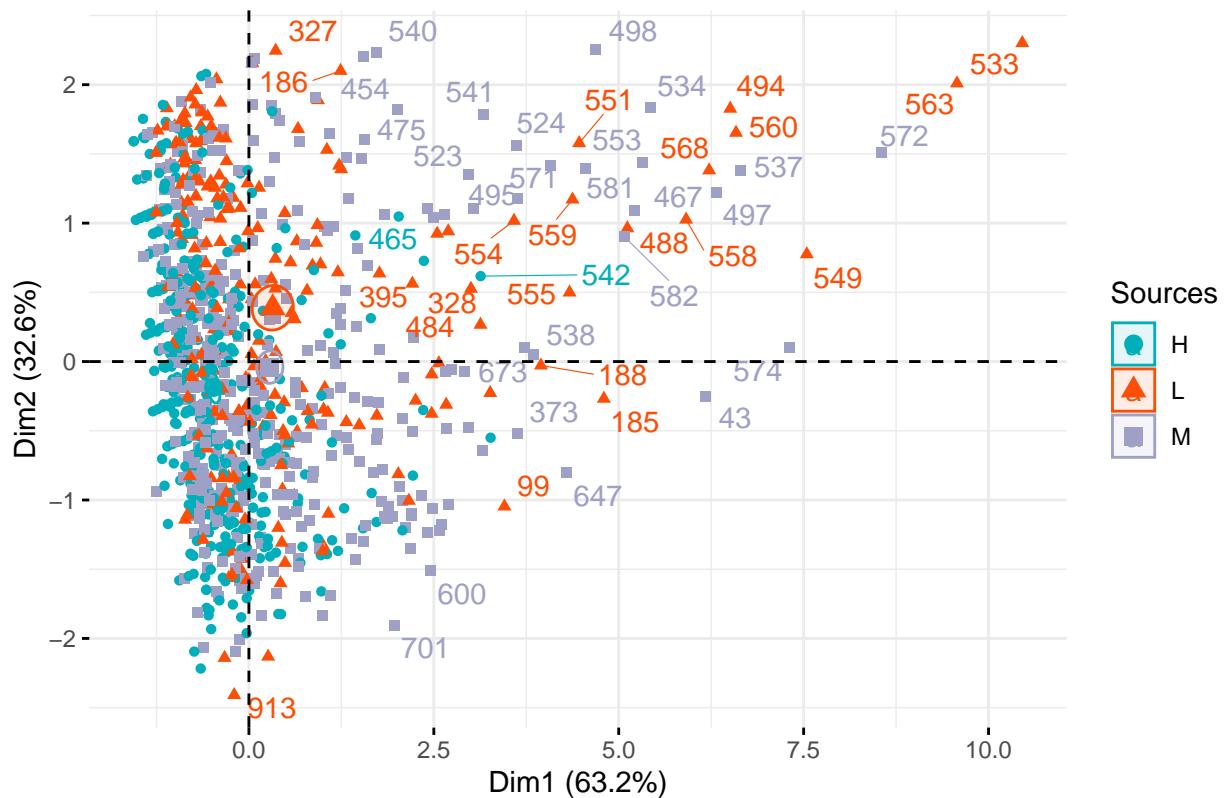
```



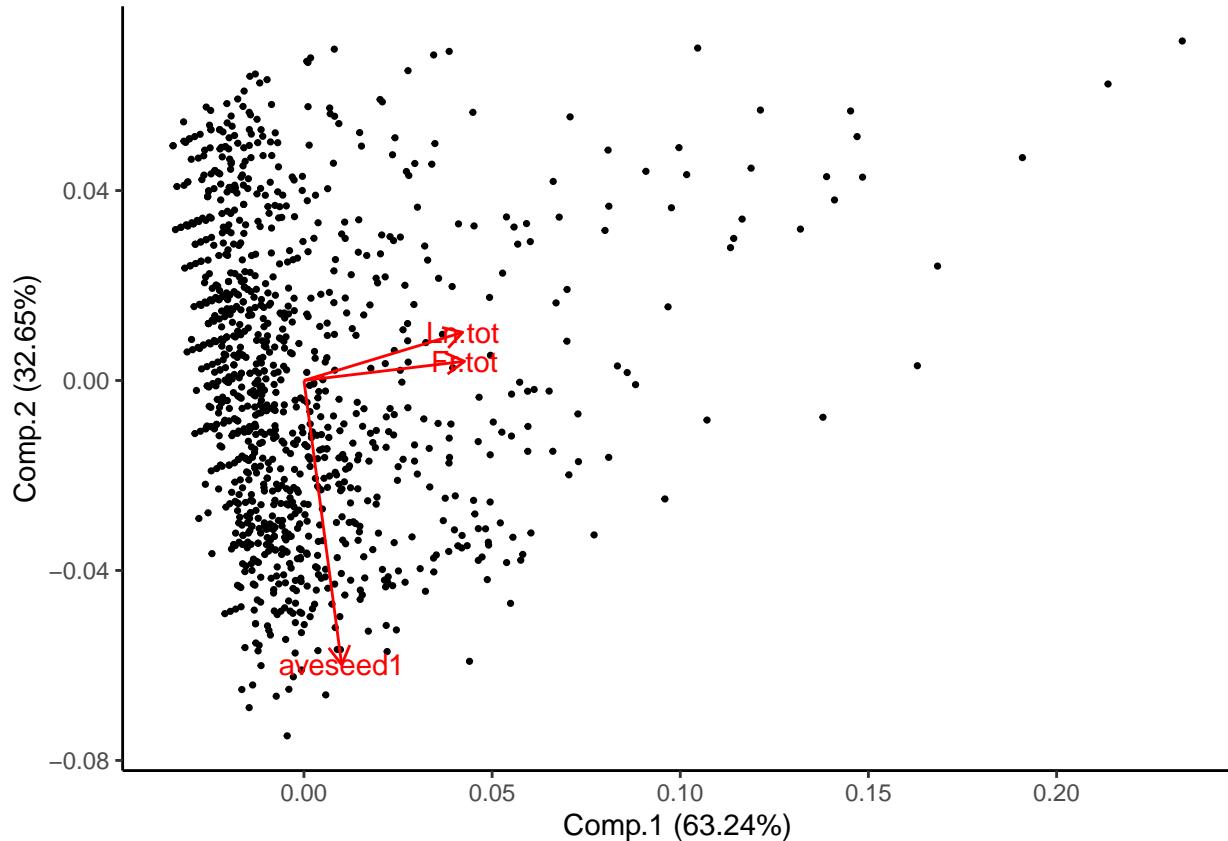
```
fviz_pca_ind(ltrsfull,
  col.ind = ltrs_pc$Source, # color by groups
  palette = c("#00AFBB", "#FC4E07", "#9FA0C5"),
  addEllipses = TRUE, # Concentration ellipses
  ellipse.type = "confidence",
  legend.title = "Sources",
  repel = TRUE
)
```

```
## Warning: ggrepel: 1010 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

## Individuals – PCA



```
autoplot(ltrsfull, data = ltrs_pc, size = 0.6, loadings = T, loadings.label = T) + theme_classic()
```



Compare the direction of these projected eigenvectors to the loadings in the output, it shows that component 1 is affected by total leaf nodes and total fruits from all stems, while component 2 is more affected by average seeds produced by the primary fruit.

## Plotting Factor Analysis

```
ltrsdat$year <- as.factor(ltrsdat$year)
ltrsdat$transect <- as.factor(ltrsdat$transect)
ltrsdat$site <- as.factor(ltrsdat$site)
ltrsdat$treat <- as.factor(ltrsdat$treat)
ltrsdat$Source <- as.factor(ltrsdat$Source)
ltrsdat$tbo <- as.factor(ltrsdat$tbo)
```

## Calculate Principal Components

```
library(FactoMineR)

## FAMD
ltrs_fa <- subset(ltrsdat, subset = site %in% c("L", "M", "H"),
                    select=-c(aveseed2, totbadseed))

ltrs_fa <- ltrs_fa %>%
  na.omit()

ltrs.famd <- FAMD(ltrs_fa,
                     sup.var = 4, ## Set the target variable "treatment" as a supplementary variable
```

```

graph = FALSE,
ncp=25)

## Inspect principal components
get_eigenvalue(ltrs.famd)

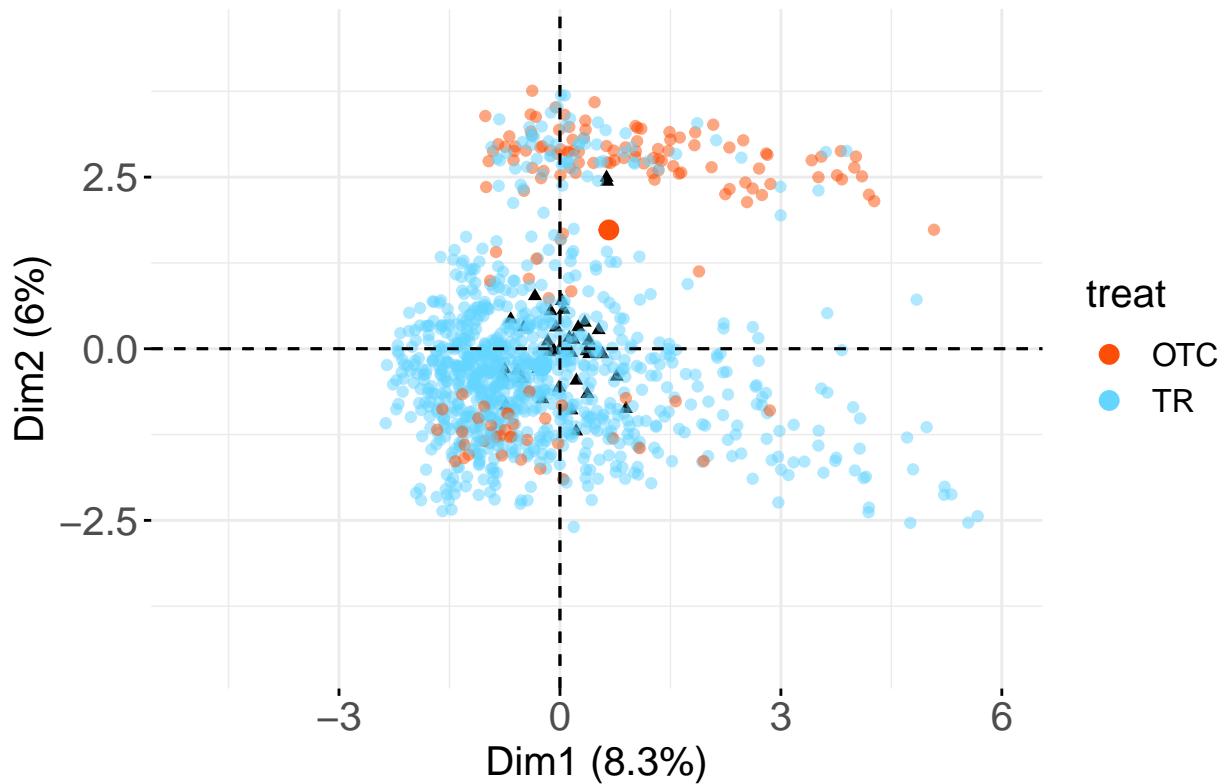
##      eigenvalue variance.percent cumulative.variance.percent
## Dim.1    3.0762190     8.314105          8.314105
## Dim.2    2.2098516     5.972572         14.286677
## Dim.3    1.3688302     3.699541         17.986218
## Dim.4    1.2418825     3.356439         21.342657
## Dim.5    1.1834416     3.198491         24.541148
## Dim.6    1.1310814     3.056977         27.598125
## Dim.7    1.0973098     2.965702         30.563827
## Dim.8    1.0474033     2.830820         33.394647
## Dim.9    1.0233734     2.765874         36.160521
## Dim.10   1.0103507     2.730677         38.891199
## Dim.11   1.0022219     2.708708         41.599906
## Dim.12   1.0017505     2.707434         44.307340
## Dim.13   1.0004528     2.703927         47.011267
## Dim.14   1.0000000     2.702703         49.713970
## Dim.15   1.0000000     2.702703         52.416672
## Dim.16   1.0000000     2.702703         55.119375
## Dim.17   1.0000000     2.702703         57.822078
## Dim.18   1.0000000     2.702703         60.524780
## Dim.19   1.0000000     2.702703         63.227483
## Dim.20   1.0000000     2.702703         65.930186
## Dim.21   1.0000000     2.702703         68.632888
## Dim.22   1.0000000     2.702703         71.335591
## Dim.23   1.0000000     2.702703         74.038294
## Dim.24   1.0000000     2.702703         76.740997
## Dim.25   0.9842831     2.660224         79.401221

## Plot individual observations
fviz_famd_ind(ltrs.famd, label = "none",
               habillage = "treat", palette = c("#FC4E07", "#63d5ff"), # color by groups
               repel = TRUE, alpha.ind = 0.5) +
  xlim(-5, 6) + ylim (-4.5, 4.5) +
  theme(text = element_text(size = 15), axis.text.x = element_text(size = 15), axis.text.y = element_text(size = 15))

## Warning: Removed 17 rows containing non-finite values (stat_mean).
## Warning: Removed 1 rows containing missing values (geom_point).
## Warning: Removed 17 rows containing missing values (geom_point).

```

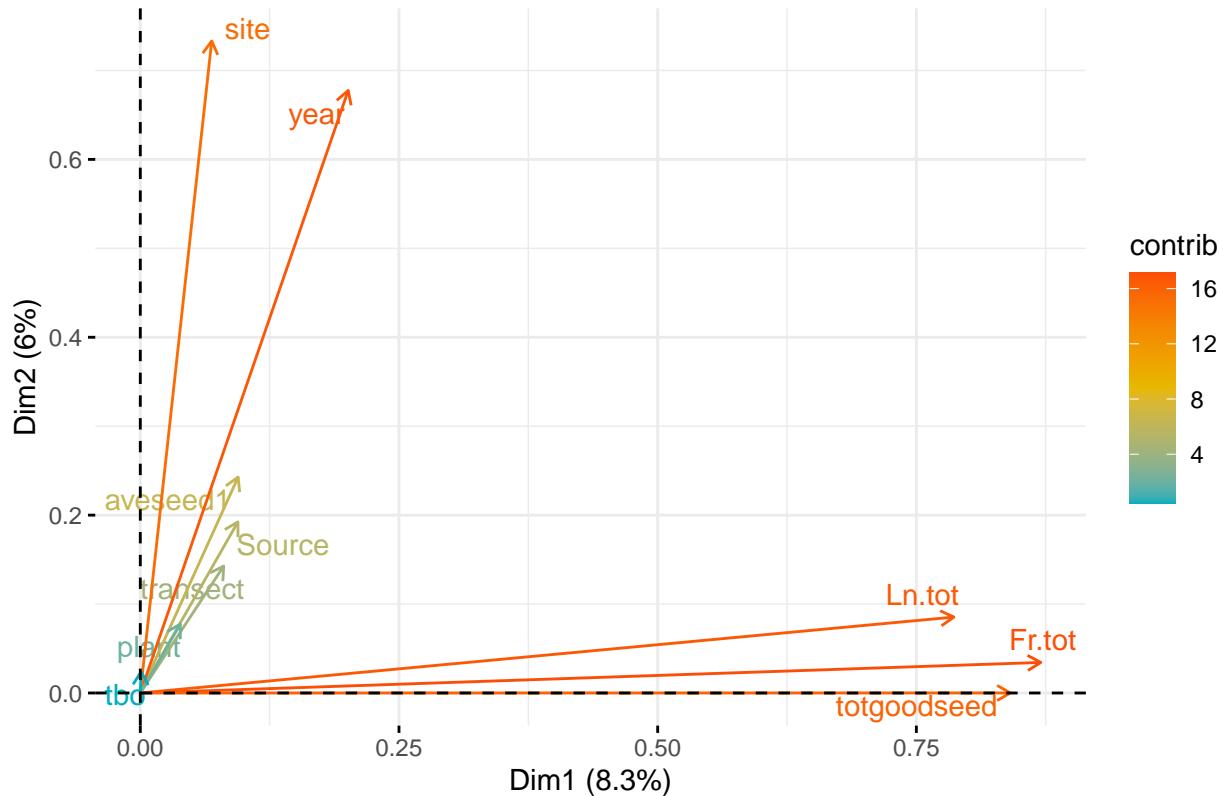
## Individuals – FAMD



As individuals with similar characteristics (in this case, in terms of plants' before- and after- habitat and reproductive success) are close to each other on the figure, the large overlap between the "OTC" and "TR" populations suggests that if there are significant/meaningful differences between control and warmed groups, they are likely complex and non-linear.

```
#visualize the graph of variables
fviz_pca_var(ltrs.famd,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE      # Avoid text overlapping
             )
```

## Variables – FAMD



Positive correlated variables point to the same side of the plot. Negative correlated variables point to opposite sides of the graph.

## Seed emergence vs. Range Analysis

Question: do plants transplanted to warmer ranged seed earlier than plants in cooler ranges, and does this change between NK and HB?

Potential relationship between seed emergence(nEmzt) and range of plant habitats.

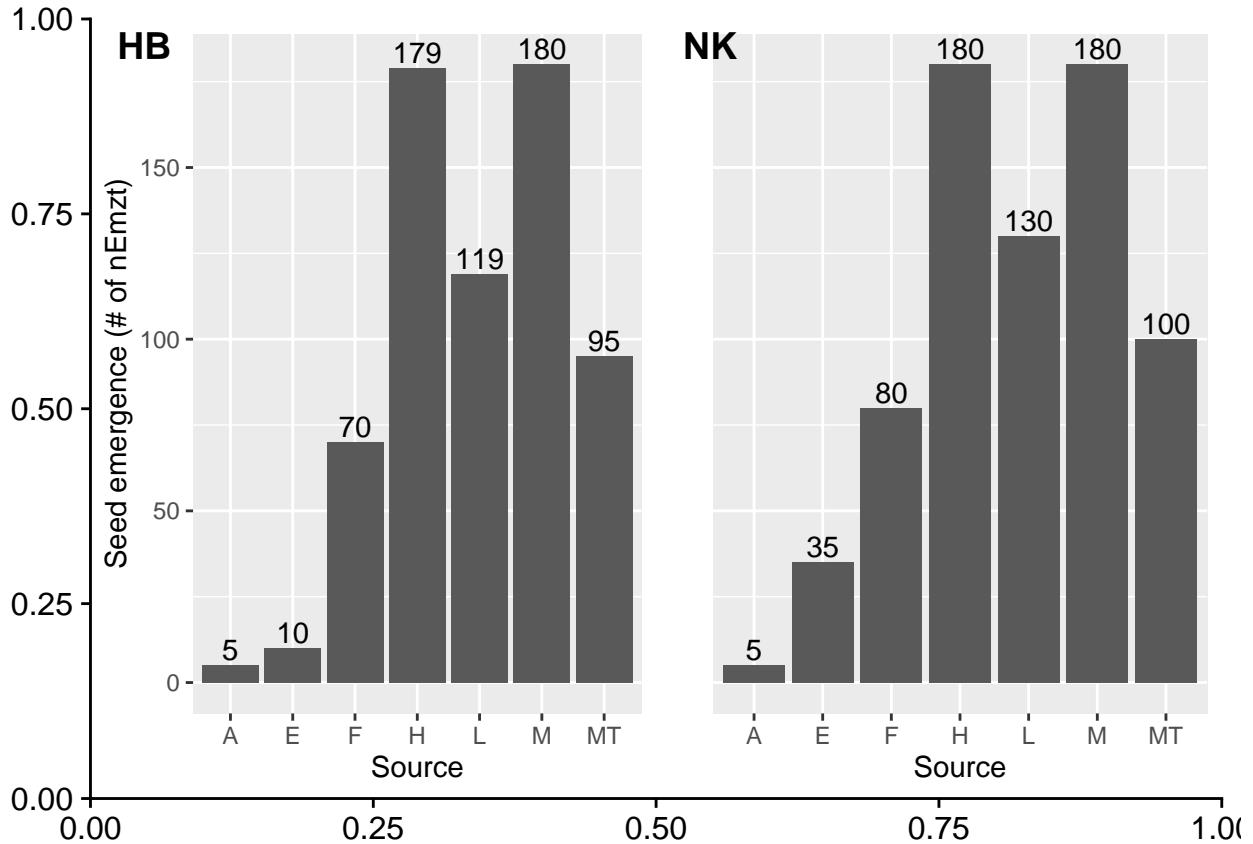
```
#plot for HB created
proHB_1 <- ggplot(proHB,
  aes(x = Source,
      fill = nEmzt)) +
  geom_bar(position = position_dodge(width=0.8)) +
  geom_text(aes(label=..count..), stat= "count", position=position_dodge(width=0.9), vjust=-0.25) +
  ylab("Seed emergence (# of nEmzt)")

#plot for NK created
proNK_1 <- ggplot(proNK,
  aes(x = Source,
      fill = nEmzt)) +
  geom_bar(position = position_dodge(width=0.8))+
```

```

geom_text(aes(label=..count..), stat= "count", position=position_dodge(width=0.9), vjust=-0.25)+  
ylab("")+  
theme(axis.text.y=element_blank(),  
axis.ticks.y=element_blank())  
  
#the above two plots are arranged side-to-side for the purpose of comparison  
plot_grid(proHB_1, proNK_1, labels = c('HB','NK'), ncol=2) + theme_cowplot()

```



As seen in the plot, there is an obvious increase in the number of plant emergence at source E, which stands for the absolute high-elevation range edge, in NK comparing to HB. Increase in plant emergence are also seen in F (Fortress Mountain), L (low-elevation from that transect), and MT (neighbouring Moose Mountain above 2000m).

## Group by transplant location

In this section, groups are categorized by transplant location (“treat”, control or warming chambers) using similar codes above.

```

proHB_2 <- ggplot(proHB, aes(treat))+  
  geom_bar() +  
  scale_y_continuous(limits=c(0, 600)) +  
  geom_text(aes(label=..count..), stat= "count", position=position_dodge(width=0.9), vjust=-0.25)  
  
proNK_2 <- ggplot(proNK, aes(treat))+  
  geom_bar() +  
  ylab("") +

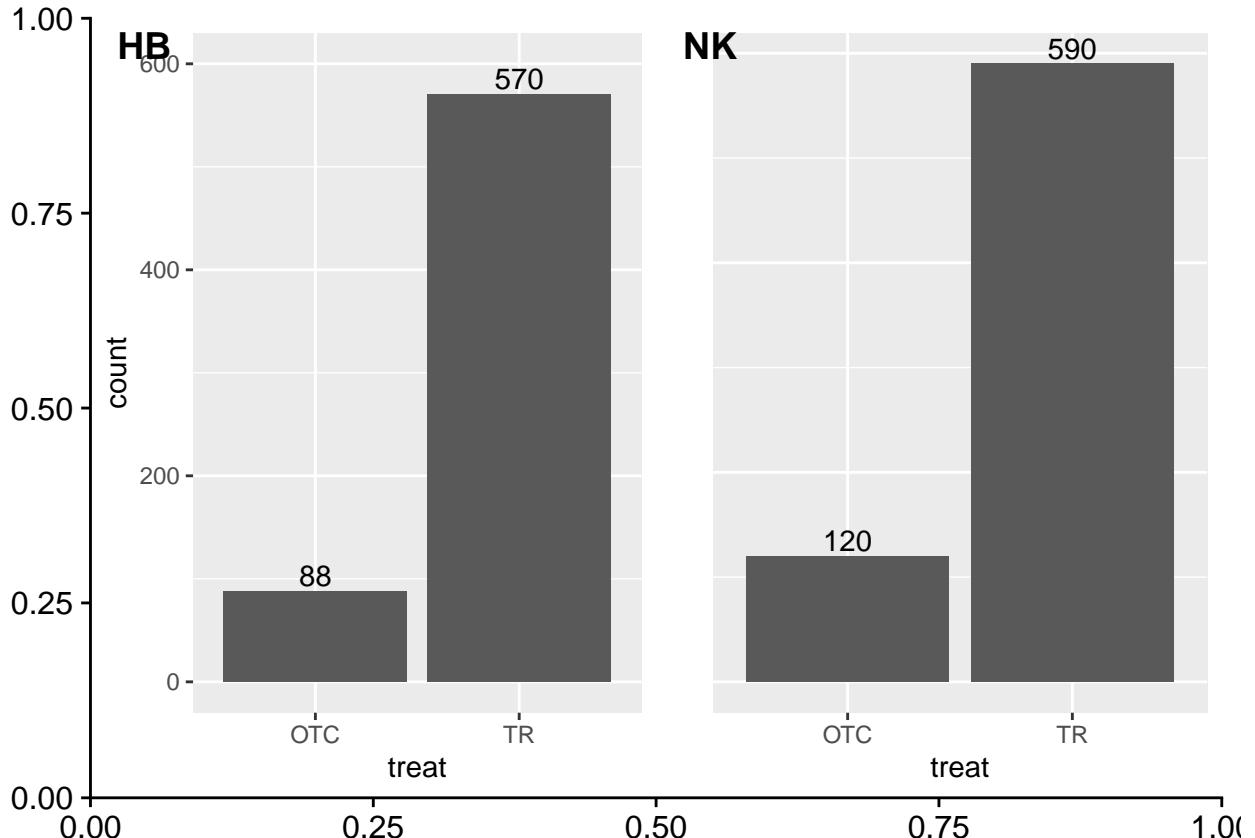
```

```

geom_text(aes(label=..count..), stat= "count", position=position_dodge(width=0.9), vjust=-0.25) +
theme(axis.text.y=element_blank(),
axis.ticks.y=element_blank())

plot_grid(proHB_2, proNK_2, labels = c('HB','NK'), ncol=2) + theme_cowplot()

```



According to plots generated above indicates that the controlled normal plant population (TR) has greater number comparing to the ones within the Open topped warming chambers (OTC). When comparing the count number between HB and NK, both treats increase by approximately 40. Therefore, it is predicted that plants from higher elevations transplanted to warmer ranges will seed earlier. Plants from lower elevations transplanted to cooler ranges will seed later.