R_markdown_iLand_exploratory_analysis.r

baldo

2023-06-14

```
# R wind plot events random - paper on recovery rate after disturbance (09.11.2022)

# R wind plot event in block - paper on recovery rate after disturbance (before editing)


library(RSQLite)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(fields)
```

```
## Loading required package: spam
```

```
## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Warning: package 'viridis' was built under R version 4.2.3

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
# Path to search the data
dataroot <- ("C:/iLand/2023/browsing_revision/db_test/")             # Root for the selection of the da

# CREATE NEW EMPTY DATAFRAME                                                 # this is needed to cre

# removals<-c()

lnd<-c()

aUnit<-c()

bb <-c()

w <- c()

damage.all<-c()

dys <- c()

variables.all <- c()
```

```
# FOR CYCLE FOR THE IMPORT AND ANALYSIS OF THE LANDSCAPE VOLUME AND VOLUME HARVESTED

all_v <- list.files(dataroot, ".sqlite")                              # alternative way to select all t


for (i in (1:length(all_v)))  {                                       # We read in the files in the loo

  # "for" = for argument, "(in"= in, "1"= first element of the for cycle to analysis, ": length(cases))
  # PAY ATTENTION this part HERE is just FOR TESTING

  #i<-1 # to test but remember to don't run also the }

  # ORDINATION OF THE CASE TO IMPORT AS DATABASE see line 45 the loop will be as long as number of case
```

```r
# Name of the database
file <-paste0(dataroot, all_v[i])                                    # File to read here the case is
# "file"= name of the object, "paste0"+ function to create a NAME for a computer path of selection of

# case<- strsplit(all_v[i],".s")[[1]][1]      # Why we used it in the paper analysis? # explanation o
case<-all_v[i]


# Control
print(file)

# connect to the database of clear cut model
sqlite.driver <- dbDriver("SQLite")
db1 <- dbConnect(sqlite.driver, dbname = file)  # connect to the file
tables.in.the.file<-dbListTables(db1)           # explore the tables in the file
print(tables.in.the.file)


#-------------------------------------------------------------------------------
# LOAD THE DATABASE

landscape <- dbReadTable(db1,"landscape")
abeUnit <- dbReadTable(db1, "abeUnit")
#abeStandRemoval <- dbReadTable(db1,"abeStandRemoval")
barkbeetle <- dbReadTable(db1,"barkbeetle")
wind <- dbReadTable(db1,"wind")
dynamicstand <- dbReadTable(db1,"dynamicstand")
carbon <- dbReadTable(db1,"carbon")
carbonflow <- dbReadTable(db1, "carbonflow")

dbDisconnect(db1)    # close the file


#-------------------------------------------------------------------------------
# CREATE SHANNON VARIABLE

annual.data<-landscape %>% group_by(year) %>% filter(year>0) %>% summarize(VOL.tot=sum(volume_m3), BA
annual.spec.data<-landscape %>% group_by(year, species) %>%  filter(year>0) %>% summarize(VOL=(volume_

print(head(annual.data))
print(head(annual.spec.data))

S<-landscape %>% group_by(year) %>% filter(volume_m3>0 & year>0) %>% summarise(n=n())    # number of s
print(S$n)


t<-annual.spec.data %>% right_join(annual.data, by="year")  %>%
  mutate(prop.VOL=VOL/VOL.tot, prop.BA=BA/BA.tot, prop.count=count/count.tot) %>%
  filter(prop.VOL>0)  # here I also filtering them to have only records with m3>0

#https://www.statology.org/shannon-diversity-index/
#https://www.statology.org/shannon-diversity-index-calculator/
```

```r
# Shannon diversity index (SDI): this is already Shannon.... that extra step by dividing the by S$n i
# so maybe we can make shannon based on BA, VOL and number of trees.... (? can discuss or save all 3

H.BA<-t %>% group_by(year) %>% summarize(H=-1*sum(prop.BA*log(prop.BA)))
H.VOL<-t %>% group_by(year) %>% summarize(H=-1*sum(prop.VOL*log(prop.VOL)))
H.count<-t %>% group_by(year) %>% summarize(H=-1*sum(prop.count*log(prop.count)))   # here I just put
set.panel(3,1)
plot(H.BA, main= "H.BA")
plot(H.VOL, main= "H.vol")
plot(H.count, main="H.count")

#The higher the value of H, the higher the diversity of species in a particular community.
#The lower the value of H, the lower the diversity.
#A value of H = 0 indicates a community that only has one species.


# Shannon Equitability Index (SEI).... maybe we do not need this at all.....
SEI<-H.BA$H/log(S$n)
SEI[which(is.na(SEI)==T)]<-0

#-------------------------------------------------------------------------------
# Pioneer species proportion based on BA
# CSP conifer species proportion

conifers <-c("pini", "pisy", "psme", "lade", "piab", "abal")

conif.spec.prop<-t %>% filter(species %in% conifers) %>% summarize(BA=sum(prop.BA))

csp<-data.frame(year=c(1:max(landscape$year)))
csp<-csp %>% left_join(conif.spec.prop, by="year")
csp$BA[which(is.na(csp$BA)==T)]<-0



# ESP

earlyspecs<-c("lade","acps","frex","cabe","bepe","alin","algl","acca","acpl","soau","soar","coav","al

early.spec.prop<-t %>% filter(species %in% earlyspecs) %>% summarize(BA=sum(prop.BA))

esp<-data.frame(year=c(1:max(landscape$year)))
esp<-esp %>% left_join(early.spec.prop, by="year")
esp$BA[which(is.na(esp$BA)==T)]<-0

#-------------------------------------------------------------------------------
# CREATE THE DATA FRAME FOR ADD VARIABLES ABOUT CARBON IN THE FINAL DATA FRAME

ab.tot.c<- data.frame(carbon %>%
                        group_by(year) %>%
                        summarise(tot_carbon=sum(stem_c, branch_c, foliage_c, coarseRoot_c, fineRoot_

#-------------------------------------------------------------------------------
# CREATE NEE VARIABLE
```

```r
ab.nee.v<- data.frame(carbonflow %>%
                        group_by(year) %>%
                        summarise(NEE= (-1*(NPP + Rh))))


#-------------------------------------------------------------------------
# CREATE THE CALCULATION FOR DAMAGES LOOK LINE 370

landscape.area<-landscape$area[1]                                          # CREATE THE VARIABLE F

lnd_volume = landscape %>% group_by(year)  %>%                             # CREATE THE SUMMARIZAT
  summarise(tot_vol = sum(volume_m3),
            .groups = 'drop')

head(lnd_volume)

# WIND AND BARKBEETLE MERGING

head(barkbeetle)
head(wind)

damage <- data.frame(year=barkbeetle$year,                                 # CREATE THE DATA FRAME
                     barkbeetle=barkbeetle$killedVolume)

# ADD WIND IMPACT IN THE DAMAGE DATA FRAME                                 # LEFT_JOIN IS A FUNCTI
damage<-left_join(damage,wind[,c(1,8)],by=("year"))                        # NB ...wind[,c(1,8)]..
damage<-left_join(damage,lnd_volume,by=("year"))                           # ADD THE LANDSCAPE VOL


# CREATE VARIABLES AND ADD NEW COLUMNS: IMPACT (BB+WIND), IMPACT % ON TOTAL VOLUME.

# damage$killedVolume[which(is.na(damage$killedVolume)==TRUE)] <- 0          # FOR MAKE THE na = 0

# damage$wind[which(is.na(damage$wind)==TRUE)] <-0                           # FOR MAKE THE na = 0

# TOTAL IMPACT IN M3/ha
impact <- data.frame(year=damage[,1],
                     impact= (damage[,2] + damage[,4])/landscape.area)     # "impact <- data.frame
# IT IS THE SAME CODE FOR CREATE THE DATAFRAME IMPACT variable
# Add the variable
damage <- left_join(damage,impact,by=("year"))


# IMPACT RELATIVE (%) AT THE TOTAL VOLUME
rel_imp <- data.frame(year=damage$year,
                      rel_imp=((damage$impact/damage$tot_vol)*100))

damage<-left_join(damage,rel_imp,by=("year"))

# GIVE THE NAME AT EVERY VARIABLE

# colnames(damage)<-c("year","barkbeetle","case","wind","volume","impact_m3","relative_imp")
```

```r
# damage <- damage[,-3]

head(damage)


#-------------------------------------------------------------------------
# CREATE A NEW DATA FRAME TO WORK ON THE SPECIFIC VARIABLES NEEDED

head(landscape)
head(dynamicstand)

# TO UNDERSTAND THE OPERATORS %>% AND %IN% HAVE TO STUDY THEM IN DATACAMP AND IN DPLYR CRAN PACKAGES

ab.lnd.v<- data.frame(landscape %>%
                        group_by(year) %>%
                        filter(year>0) %>%
                        summarise(tot_volume=sum(volume_m3),living_c=sum(total_carbon_kg), count_ha=su




dynamicstand_1 <-dynamicstand %>% filter(year>0)                              # THIS IS NEEDED FOR MAKE

# CREATE THE NEW DATA FRAME FOR VARIABLES

variables <- data.frame(case=case,
                        year=dynamicstand_1$year,
                        h=dynamicstand_1$height_mean,
                        dbh=dynamicstand_1$dbh_mean,
                        age=dynamicstand_1$age_mean,
                        Rh=carbonflow$Rh,
                        NPP=carbonflow$NPP,
                        GPP=carbonflow$GPP,
                        NEP=carbonflow$NEP,
                        H.count=H.count$H,
                        H.VOL=H.VOL$H,
                        H.BA=H.BA$H,
                        SEI=SEI,
                        esp_BA_prop=esp$BA,
                        csp_BA_prop=csp$BA)

# ADD LANDSCAPE VARIABLES AT V DATA FRAME

variables = inner_join(variables, ab.lnd.v, by="year")
variables = inner_join(variables, ab.tot.c, by="year")
variables = inner_join(variables, ab.nee.v, by="year")
variables = inner_join(variables, damage, by="year")

# variables[which(is.na(damage$wind)==TRUE)] <-0                              # FOR MAKE THE na = 0 !

head(variables)
```

```
#-------------------------------------------------------------------------
# Make the 3 categories of removals:

# activity.names<-unique(abeStandRemoval$activity)                              # here I list all dif
# swcuts<- grepl("sw",activity.names)                                           # I look for only whi
# activity.names.sw<-activity.names[swcuts]                                     # collect the activit
# activity.names.notsw<-activity.names[!swcuts]                                 # collect the activit

# print(activity.names.sw)
# print(activity.names.notsw)

# Here I filter only the listed activity names and calculate thinning/finalcut values for every year
# (each line is per ha for a stand, so I scale with the area, sum up all the harvest on the landscape

# TO UNDERSTAND THE OPERATORS %>% AND %IN% HAVE TO STUDY THEM IN DATACAMP AND IN DPLER CRAN PACKAGES

# ab.regcuts<- data.frame(abeStandRemoval %>% filter(activity %in% activity.names.sw)    %>%
# group_by(year)   %>%   summarise(volume=sum(volumeThinning*area)/landscape.area, type="regcut", run

# ab.finalcuts<- data.frame(abeStandRemoval %>% filter(activity %in% activity.names.sw)    %>%
#  group_by(year)   %>%   summarise(volume=sum(volumeFinal*area)/landscape.area, type="finalcut", run

# ab.thinnig<- data.frame(abeStandRemoval %>% filter(activity %in% activity.names.notsw)    %>%
#   group_by(year)   %>%   summarise(volume=sum(volumeThinning*area)/landscape.area, type="thinning",

# ab.salvaged<- data.frame(abeStandRemoval %>% filter(activity %in% activity.names.sw)    %>%
#   group_by(year)   %>%   summarise(volume=sum(volumeSalvaged*area)/landscape.area, type="salvager",

# CREATE A FOR CYCLE INTO THE FOR CYCLE

# if (length(activity.names[swcuts])==0) {

#  ab.finalcuts<- data.frame(abeStandRemoval %>% filter(activity %in% activity.names.notsw)    %>%
#                            group_by(year)   %>%   summarise(volume=sum(volumeFinal*area)/landscap


# CLOSE THE FOR CYCLE


# CREATE THE VARIABLE FOR THE DIFFERENT WOOD REMOVAL ACTIVITY
#  removals<-rbind(removals,ab.regcuts,ab.finalcuts,ab.thinnig,ab.salvaged)

# Collect landscape data FOR CREATE THE VARIABLE LND FOR ALL THE RUNS
landscape<- (landscape %>% mutate(run=case))
lnd<-rbind(lnd, landscape)

# Collect abeUnit data FOR CREATE THE VARIABLE ABEUNIT FOR ALL THE RUNS
abeUnit<-(abeUnit %>% mutate(run=case))
aUnit<-rbind(aUnit, abeUnit)

# Collect barkbeetle data FOR CREATE THE VARIABLE BB FOR ALL THE RUNS
barkbeetle <-(barkbeetle %>% mutate(run=case))
bb <-rbind(bb, barkbeetle)
```

```
  # Collect wind data FOR CREATE THE VARIABLE WIND FOR ALL THE RUNS
  wind <-(wind %>% mutate(run=case))
  w <-rbind(w, wind)

  # Collect dynamicstand data FOR CREATE THE VARIABLE WIND FOR ALL THE RUNS
  dynamicstand <-(dynamicstand %>% mutate(run=case)) # consider to put dynamicstand_1 for 300 rows
  dys <-rbind(dys, dynamicstand)

  # CREATE THE VARIABLE DAMAGE FOR ALL THE RUNS
  damage <-(damage %>% mutate(case=case))
  damage.all<-rbind(damage.all, damage)                                    # PUT ALL THE DAMAGE RU

  # CREATE THE VARIABLES DF FOR ALL THE RUNS
  variables.all<-rbind(variables.all, variables)

}                                                                          # CLOSE THE WHOLE FOR C
```

```
## [1] "C:/iLand/2023/browsing_revision/db_test/B0._refclim_w1.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"
```

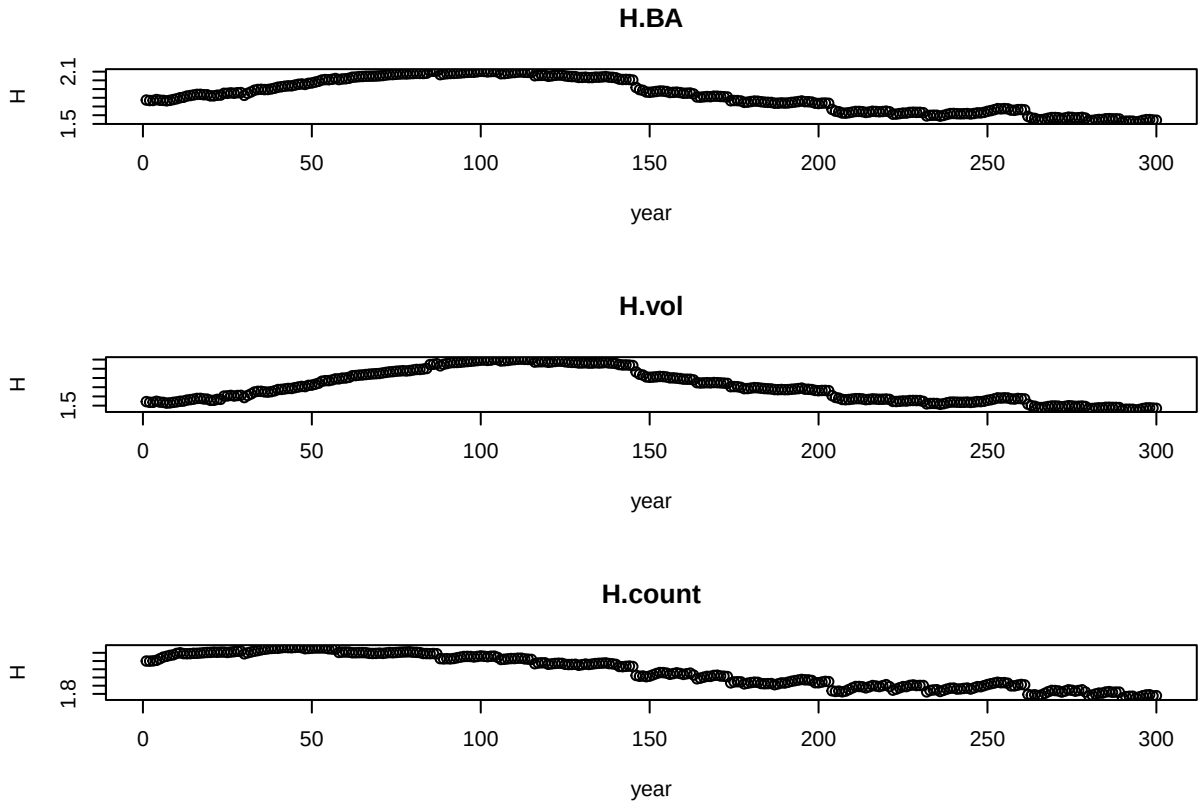
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6      999.
## 2     2    296.   30.4     1003.
## 3     3    294.   30.3     1010.
## 4     4    284.   29.3      999.
## 5     5    288.   29.9     1025.
## 6     6    297.   30.9     1055.
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.61  0.555   17.2
## 2     1 acps     7.88  1.05    67.7
## 3     1 algl     0.605 0.0859   3.57
## 4     1 bepe     0.400 0.0944   5.30
## 5     1 cabe     7.91  1.22    65.7
## 6     1 fasy    23.9   2.49   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [126] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [151] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
```

```
## [176] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [201] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [226] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 16 16 16 16 16 16
## [251] 16 16 15 15 15 15 15 15 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [276] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix
```

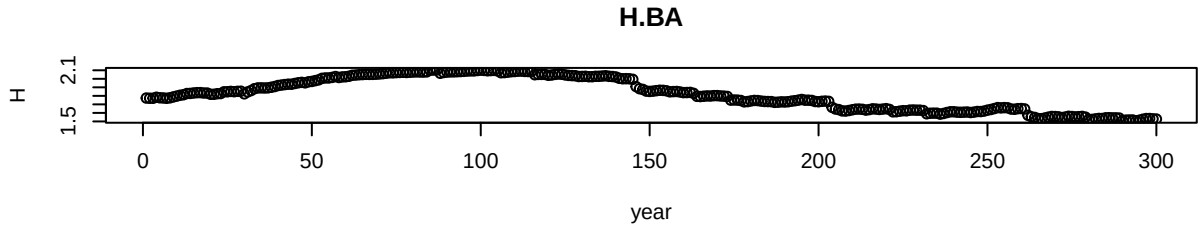
```
## [1] "C:/iLand/2023/browsing_revision/db_test/B0._refclim_w2.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"
```

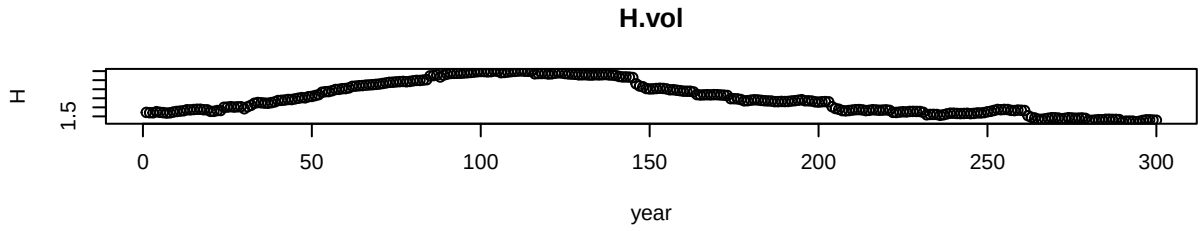
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

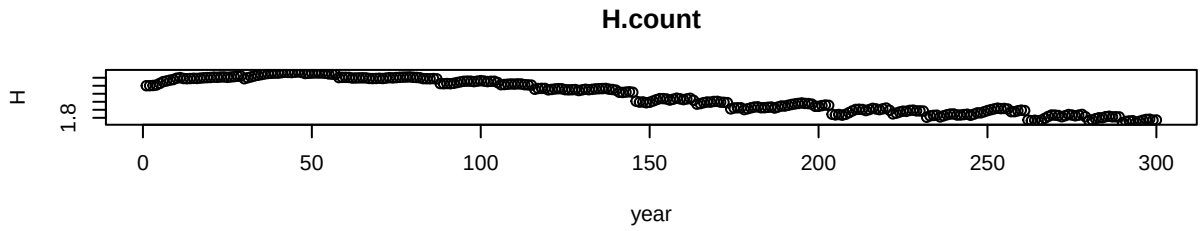



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6      999.
## 2     2    296.   30.4     1004.
## 3     3    294.   30.3     1010.
## 4     4    284.   29.4      999.
## 5     5    288.   29.9     1024.
## 6     6    297.   30.9     1054.
```

```
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.59  0.553   17.1
## 2     1 acps     7.89  1.05    67.7
## 3     1 algl     0.605 0.0860   3.57
## 4     1 bepe     0.398 0.0940   5.28
## 5     1 cabe     7.92  1.22    65.7
## 6     1 fasy    23.9   2.49   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [126] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 16 16 16 16 16 16 16 16 16 16 16
## [151] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [176] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [201] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [226] 16 16 16 16 16 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [251] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [276] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix

## [1] "C:/iLand/2023/browsing_revision/db_test/B0._refclim_w3.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"

## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```





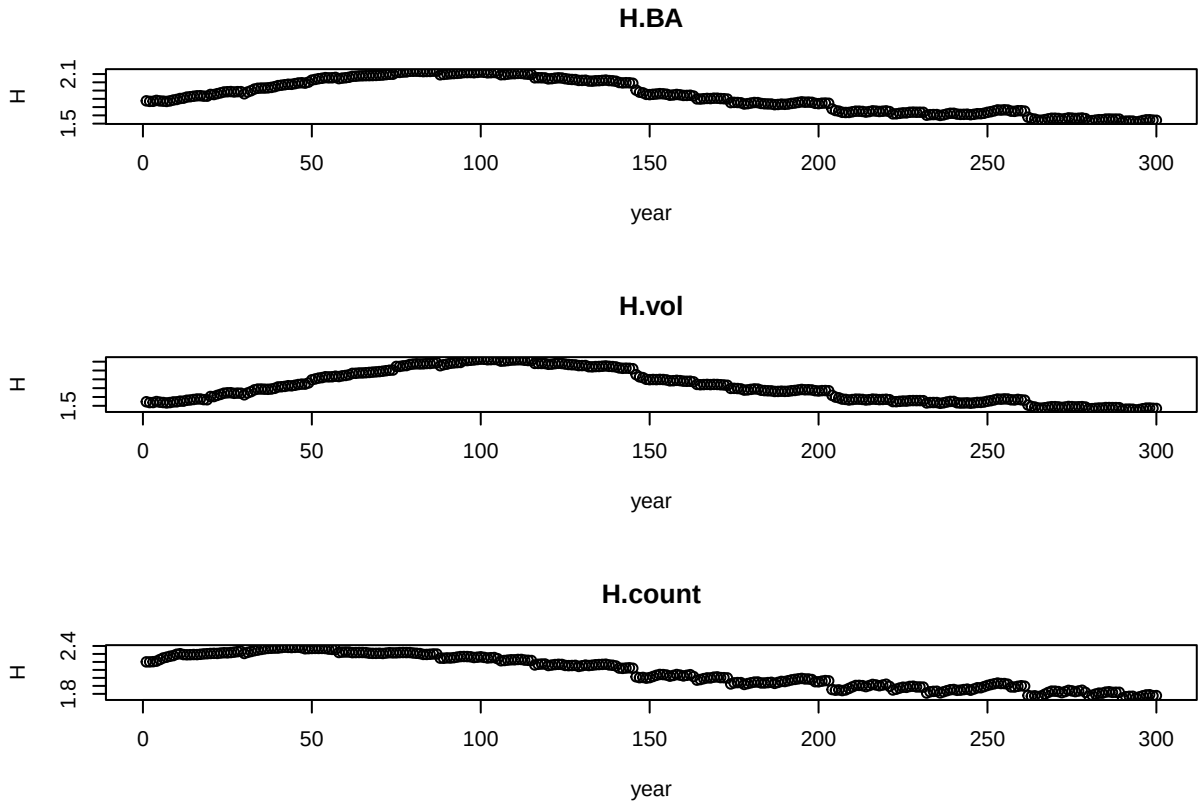



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6     1000.
## 2     2    296.   30.4     1004.
## 3     3    294.   30.3     1009.
## 4     4    284.   29.3      997.
## 5     5    288.   29.9     1022.
## 6     6    297.   30.9     1053.
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.44  0.546   17.1
## 2     1 acps     7.89  1.05    67.7
## 3     1 algl     0.604 0.0859   3.57
## 4     1 bepe     0.400 0.0944   5.30
## 5     1 cabe     7.92  1.22    65.8
## 6     1 fasy    24.1   2.51   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [126] 17 17 17 17 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [151] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
```

```
## [176] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [201] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 15 15 15
## [226] 15 16 16 16 16 16 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [251] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [276] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix
```

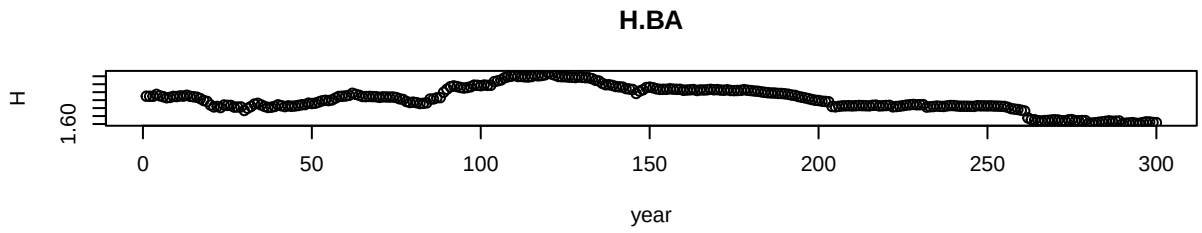
```
## [1] "C:/iLand/2023/browsing_revision/db_test/B0.5_refclim_w1.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"
```

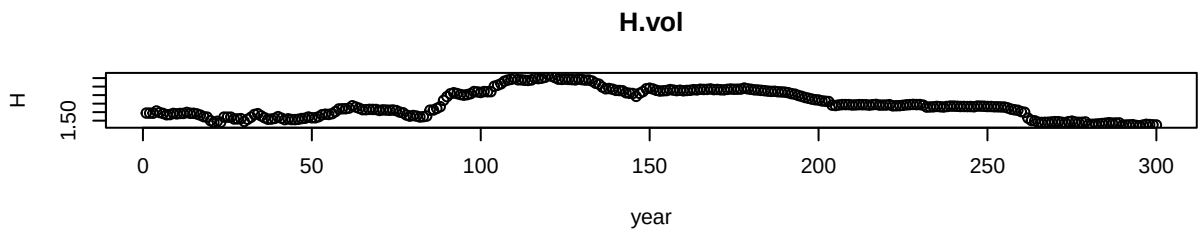
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

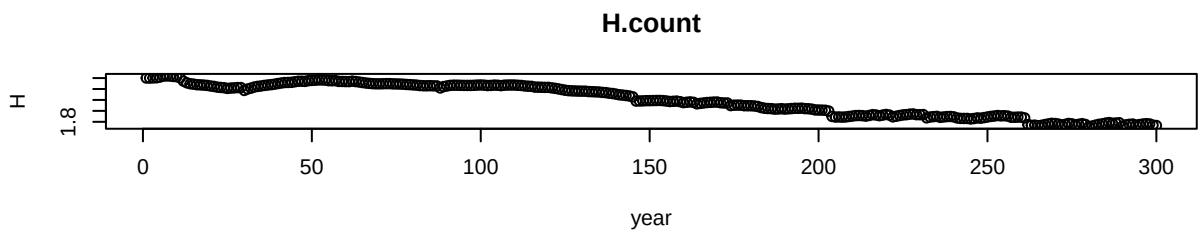



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6      999.
## 2     2    296.   30.4     1004.
## 3     3    294.   30.3     1007.
## 4     4    284.   29.4      992.
## 5     5    288.   29.8     1007.
## 6     6    297.   30.8     1024.
```

```
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.45  0.546   17.1
## 2     1 acps     7.88  1.05    67.6
## 3     1 algl     0.603 0.0857   3.57
## 4     1 bepe     0.399 0.0942   5.30
## 5     1 cabe     7.91  1.22    65.7
## 6     1 fasy    24.0   2.50   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [126] 17 17 17 17 17 17 17 17 17 17 17 17 17 16 16 16 16 16 16 16 16 16 16 16 16
## [151] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 15 15
## [176] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [201] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [226] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [251] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [276] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix

## [1] "C:/iLand/2023/browsing_revision/db_test/B0.5_refclim_w2.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"

## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```





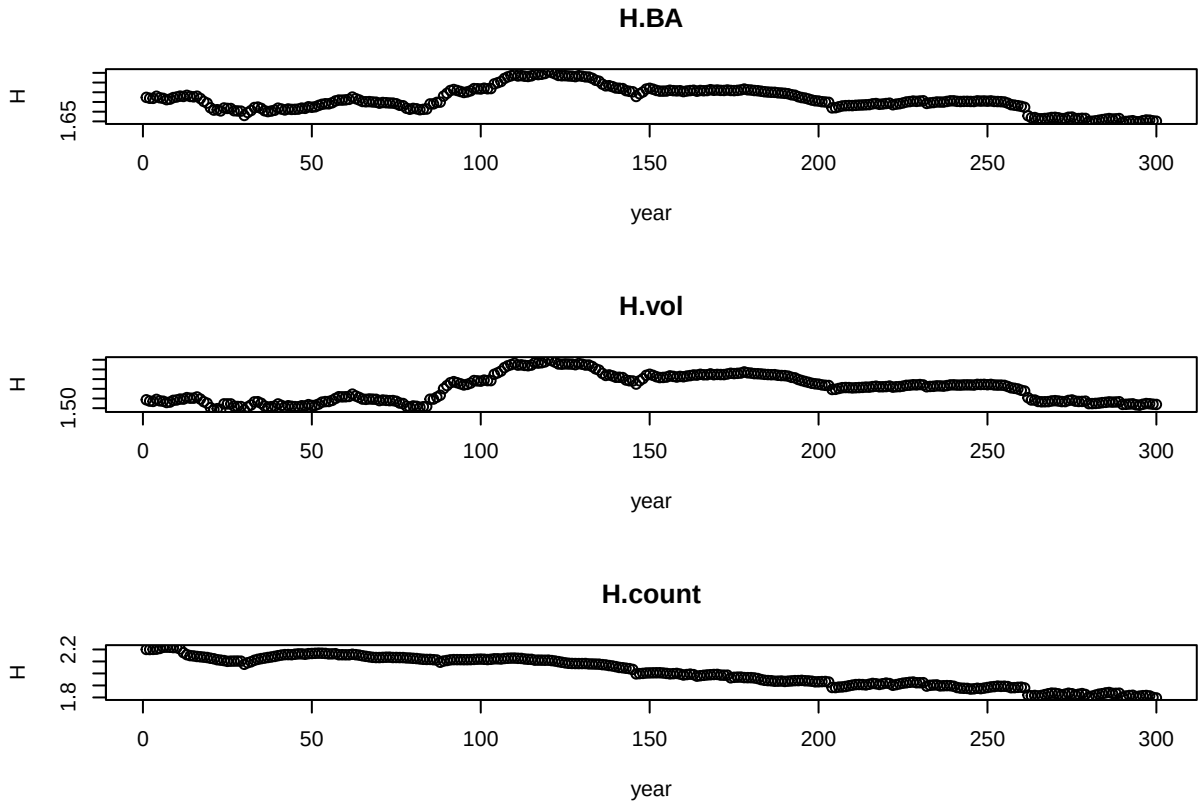



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6     1000.
## 2     2    296.   30.4     1004.
## 3     3    294.   30.3     1008.
## 4     4    284.   29.4      992.
## 5     5    288.   29.8     1007.
## 6     6    297.   30.9     1025.
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.70  0.560   17.2
## 2     1 acps     7.91  1.05    67.8
## 3     1 algl     0.605 0.0861   3.58
## 4     1 bepe     0.400 0.0945   5.30
## 5     1 cabe     7.93  1.22    65.8
## 6     1 fasy    23.4   2.46   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [126] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [151] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
```

```
## [176] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 16 16 16 15 15 15 15 15 15 15
## [201] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [226] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [251] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [276] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix
```

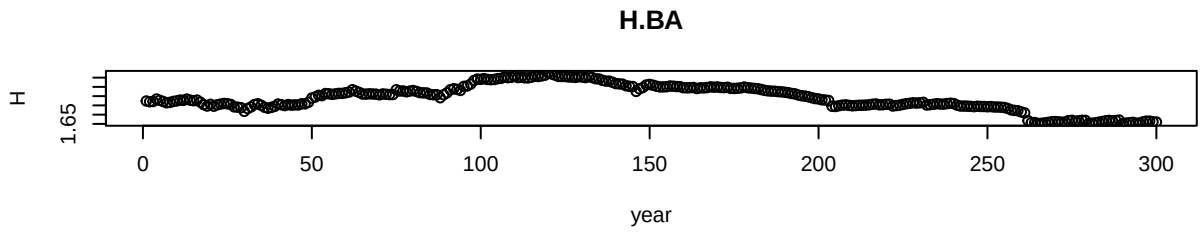
```
## [1] "C:/iLand/2023/browsing_revision/db_test/B0.5_refclim_w3.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"
```

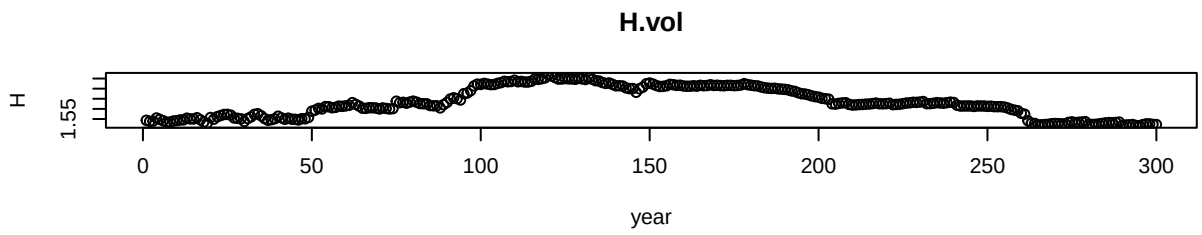
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

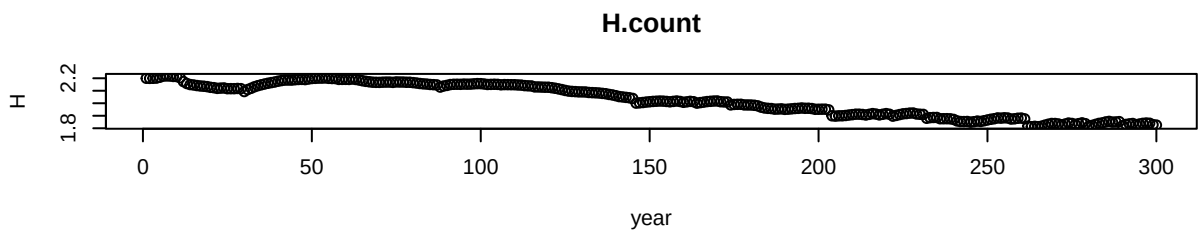



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6     1000.
## 2     2    296.   30.4     1006.
## 3     3    294.   30.3     1011.
## 4     4    284.   29.4      995.
## 5     5    288.   29.9     1009.
## 6     6    297.   30.8     1024.
```

```
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.63  0.556   17.2
## 2     1 acps     7.89  1.05    67.8
## 3     1 algl     0.604 0.0860   3.58
## 4     1 bepe     0.398 0.0940   5.28
## 5     1 cabe     7.92  1.22    65.8
## 6     1 fasy    23.9   2.49   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [126] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [151] 17 17 17 17 17 17 17 17 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [176] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 15 16 16 16 16 16 16 16 16 16 16
## [201] 16 16 16 16 16 16 16 16 16 16 16 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [226] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [251] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [276] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix

## [1] "C:/iLand/2023/browsing_revision/db_test/B1_refclim_w1.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"

## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```





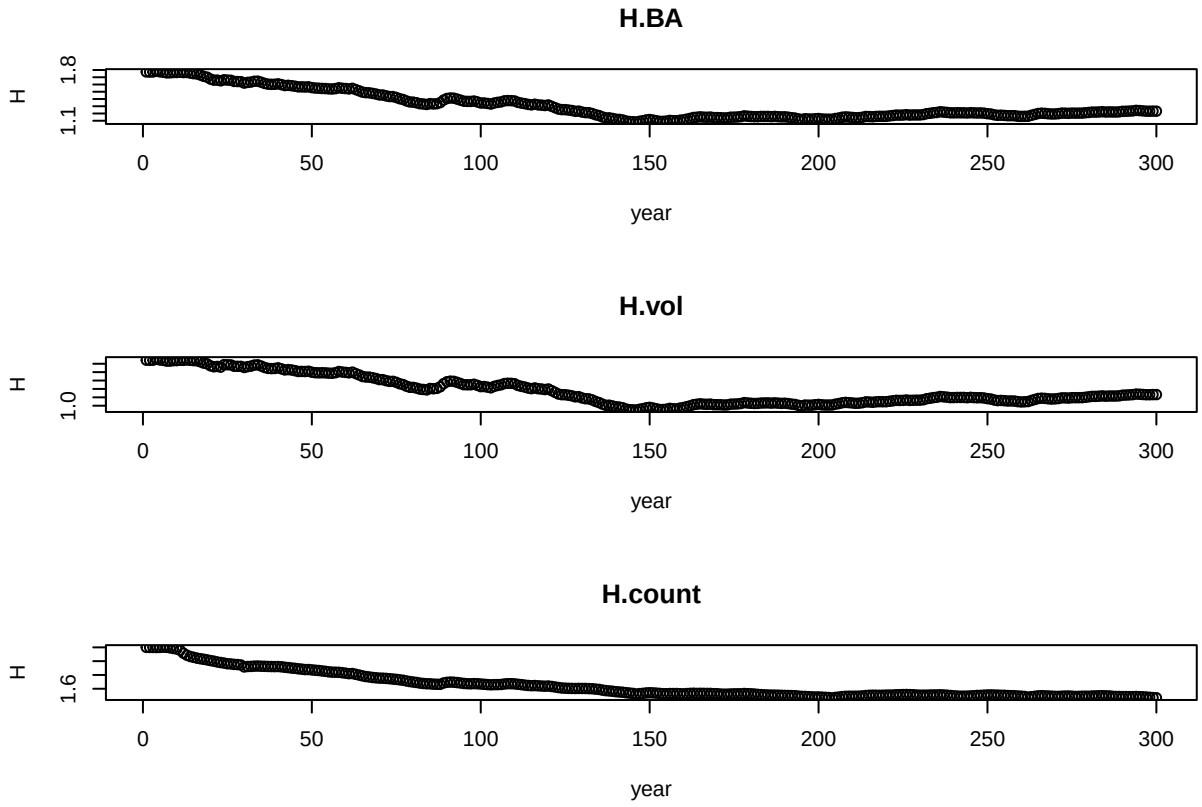



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6     1000.
## 2     2    296.   30.4     1007.
## 3     3    294.   30.3     1010.
## 4     4    284.   29.4      992.
## 5     5    288.   29.9     1003.
## 6     6    297.   30.8     1014.
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.63  0.557   17.2
## 2     1 acps     7.88  1.05    67.7
## 3     1 algl     0.604 0.0859   3.57
## 4     1 bepe     0.400 0.0945   5.30
## 5     1 cabe     7.92  1.22    65.7
## 6     1 fasy    23.6   2.47   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [126] 17 17 17 16 16 16 16 16 16 16 16 16 15 15 15 15 15 15 15 15 15 15 15 15 15
## [151] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
```

```
## [176] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [201] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [226] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [251] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [276] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix
```

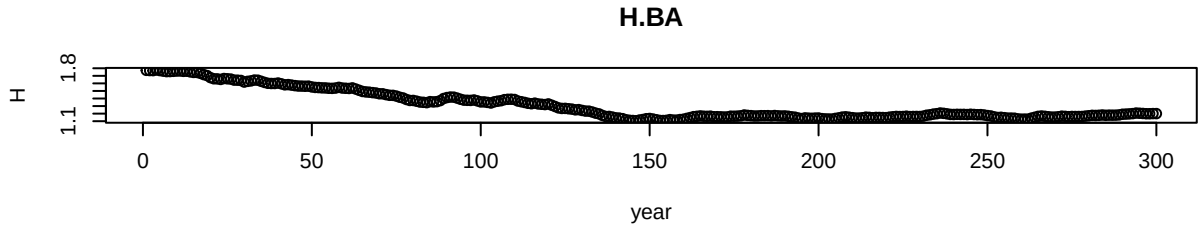
```
## [1] "C:/iLand/2023/browsing_revision/db_test/B1_refclim_w2.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"
```

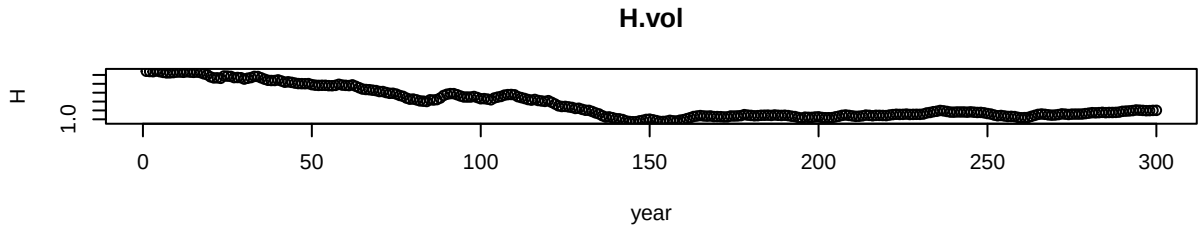
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

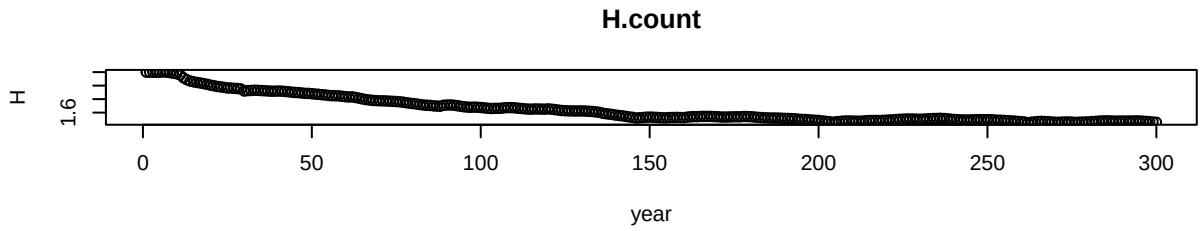



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6     1000.
## 2     2    296.   30.4     1005.
## 3     3    294.   30.3     1008.
## 4     4    284.   29.3      990.
## 5     5    288.   29.9     1003.
## 6     6    297.   30.8     1014.
```

```
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.63  0.557   17.2
## 2     1 acps     7.86  1.05    67.7
## 3     1 algl     0.605 0.0860   3.57
## 4     1 bepe     0.400 0.0945   5.30
## 5     1 cabe     7.91  1.22    65.7
## 6     1 fasy    23.6   2.47   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 16 16 16 16 16 16 16 16 16
## [126] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [151] 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 15 15 15 15 15 15 15 15
## [176] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [201] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [226] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [251] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [276] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix

## [1] "C:/iLand/2023/browsing_revision/db_test/B1_refclim_w3.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"

## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```





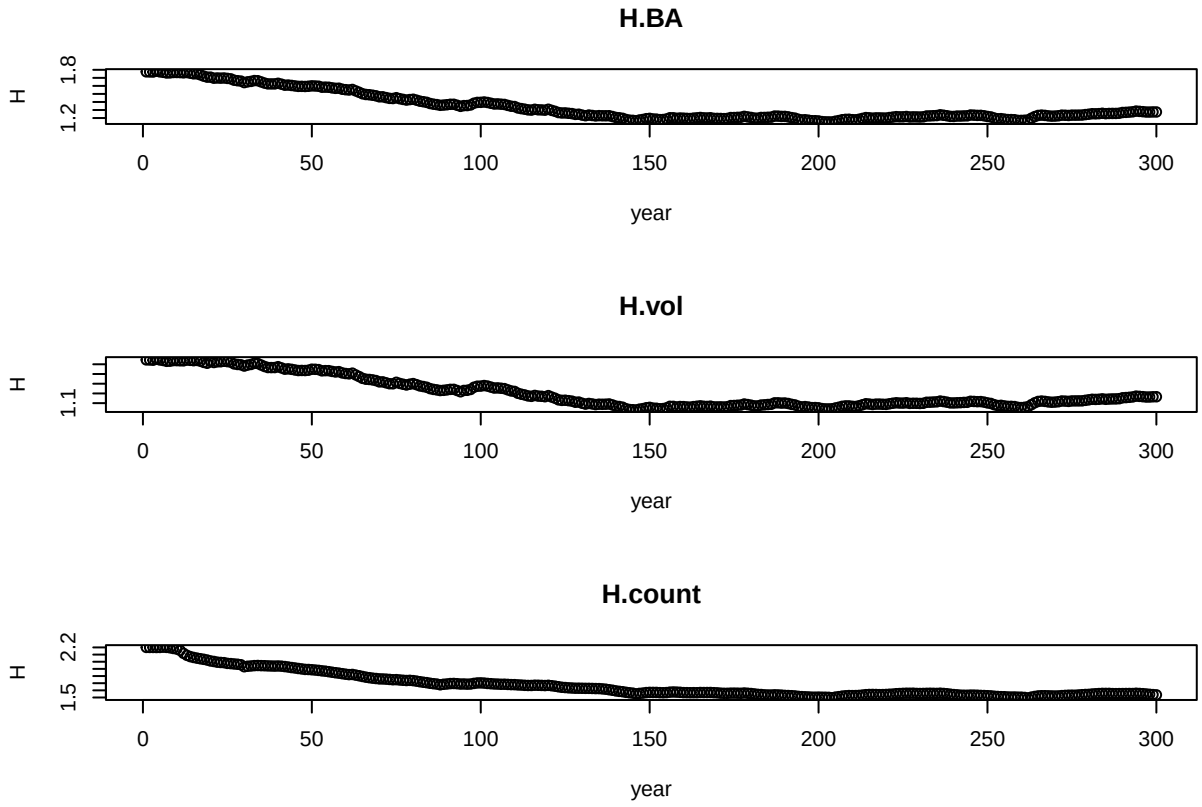



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6     1000.
## 2     2    296.   30.4     1003.
## 3     3    294.   30.3     1007.
## 4     4    284.   29.4      991.
## 5     5    288.   29.9     1002.
## 6     6    297.   30.8     1013.
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.53  0.549   17.1
## 2     1 acps     7.91  1.05    67.8
## 3     1 algl     0.606 0.0862   3.58
## 4     1 bepe     0.398 0.0940   5.28
## 5     1 cabe     7.93  1.22    65.7
## 6     1 fasy    24.0   2.50   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [101] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [126] 17 17 17 17 17 17 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 15 15 15 15
## [151] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
```

```
## [176] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [201] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [226] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [251] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [276] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## plot window will lay out plots in a 3 by 1 matrix
```

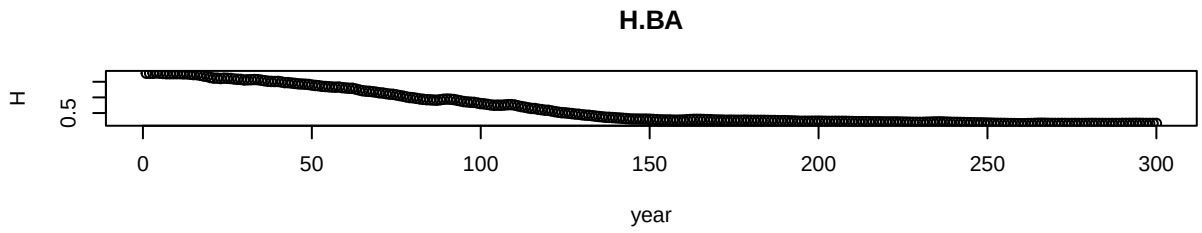
```
## [1] "C:/iLand/2023/browsing_revision/db_test/B2_refclim_w1.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"
```

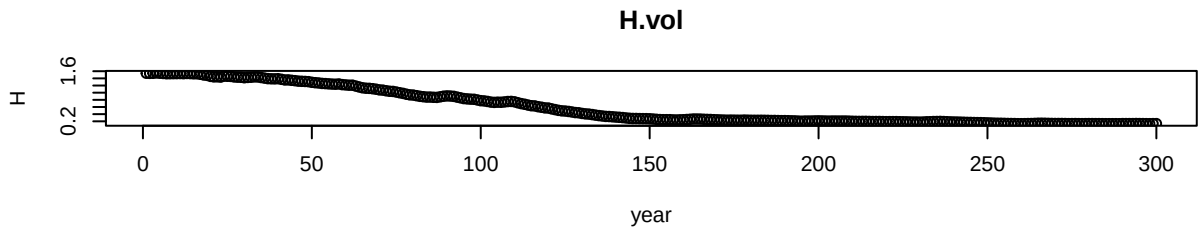
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

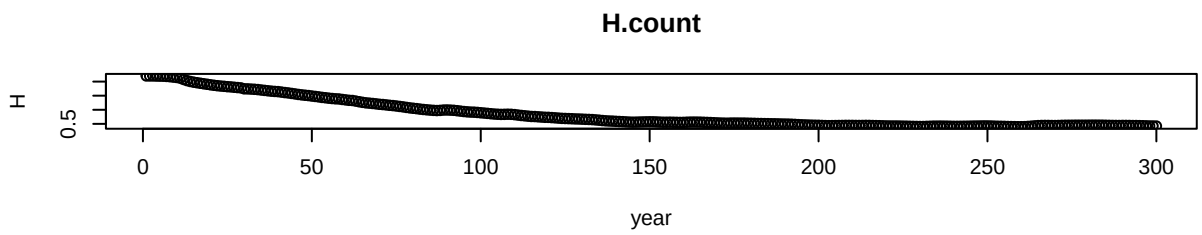



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6      999.
## 2     2    296.   30.4     1003.
## 3     3    294.   30.3     1007.
## 4     4    284.   29.4      989.
## 5     5    288.   29.9      998.
## 6     6    297.   30.9     1005.
```

```
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.54  0.551   17.1
## 2     1 acps     7.91  1.05    67.7
## 3     1 algl     0.606 0.0861   3.58
## 4     1 bepe     0.400 0.0944   5.30
## 5     1 cabe     7.92  1.22    65.7
## 6     1 fasy    23.8   2.49   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
## [101] 16 16 16 16 16 16 16 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 14 14 14
## [126] 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
## [151] 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
## [176] 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
## [201] 14 14 14 14 14 14 14 14 14 14 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
## [226] 13 13 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
## [251] 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 10 10 10 10 10 10 10
## [276] 10 10 10 10 10 10  9  9  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8
## plot window will lay out plots in a 3 by 1 matrix

## [1] "C:/iLand/2023/browsing_revision/db_test/B2_refclim_w2.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"

## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```





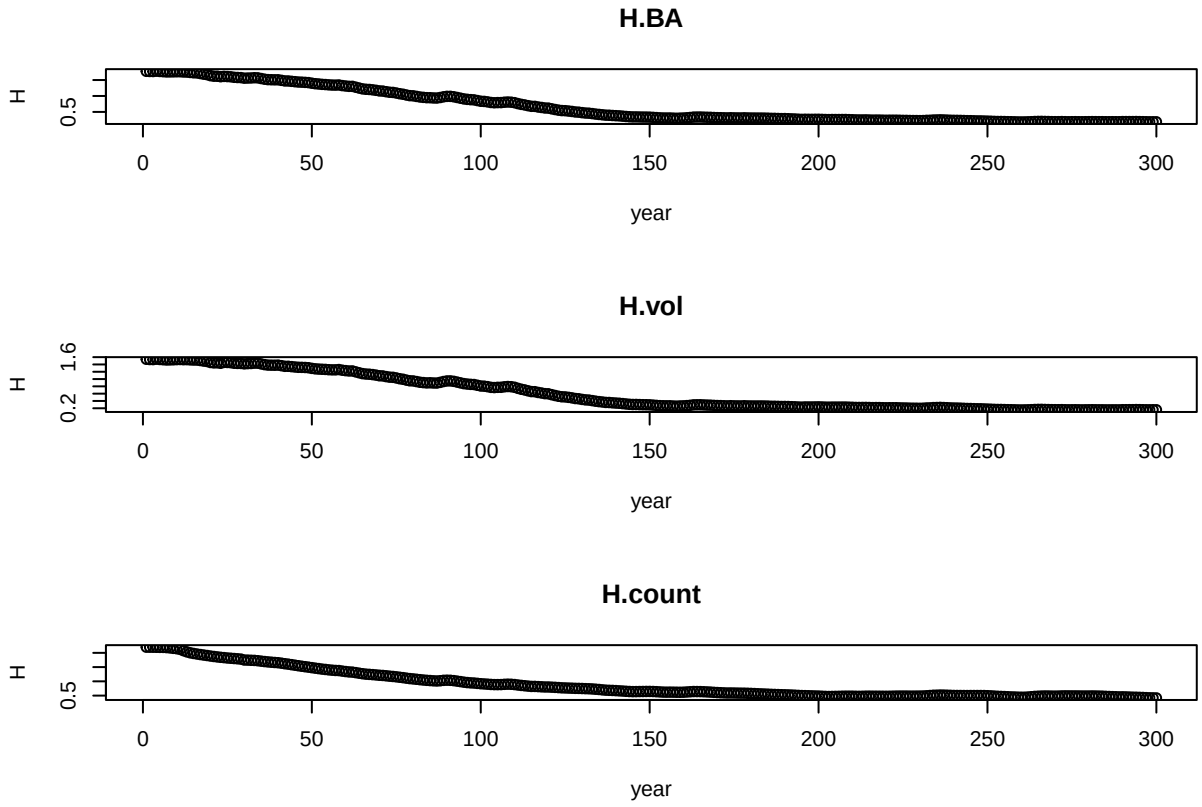



```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6      999.
## 2     2    296.   30.4     1004.
## 3     3    294.   30.3     1006.
## 4     4    284.   29.4      988.
## 5     5    288.   29.8      995.
## 6     6    297.   30.8     1001.
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.63  0.556   17.2
## 2     1 acps     7.90  1.05    67.7
## 3     1 algl     0.605 0.0861   3.58
## 4     1 bepe     0.396 0.0937   5.28
## 5     1 cabe     7.90  1.22    65.7
## 6     1 fasy    24.0   2.50   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 16 16 16 16 16 16 16 16 16 16 16 17 17
## [101] 17 17 17 17 17 16 16 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [126] 15 15 15 15 15 15 15 15 15 15 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
## [151] 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
```

```
## [176] 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
## [201] 14 14 14 14 14 14 14 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
## [226] 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 12 12 12 12
## [251] 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
## [276] 12 11 11 11 11 11 11 10 10 10 10 10 10 10  9  9  9  9  9  9  9  9  9  9  9
## plot window will lay out plots in a 3 by 1 matrix
```

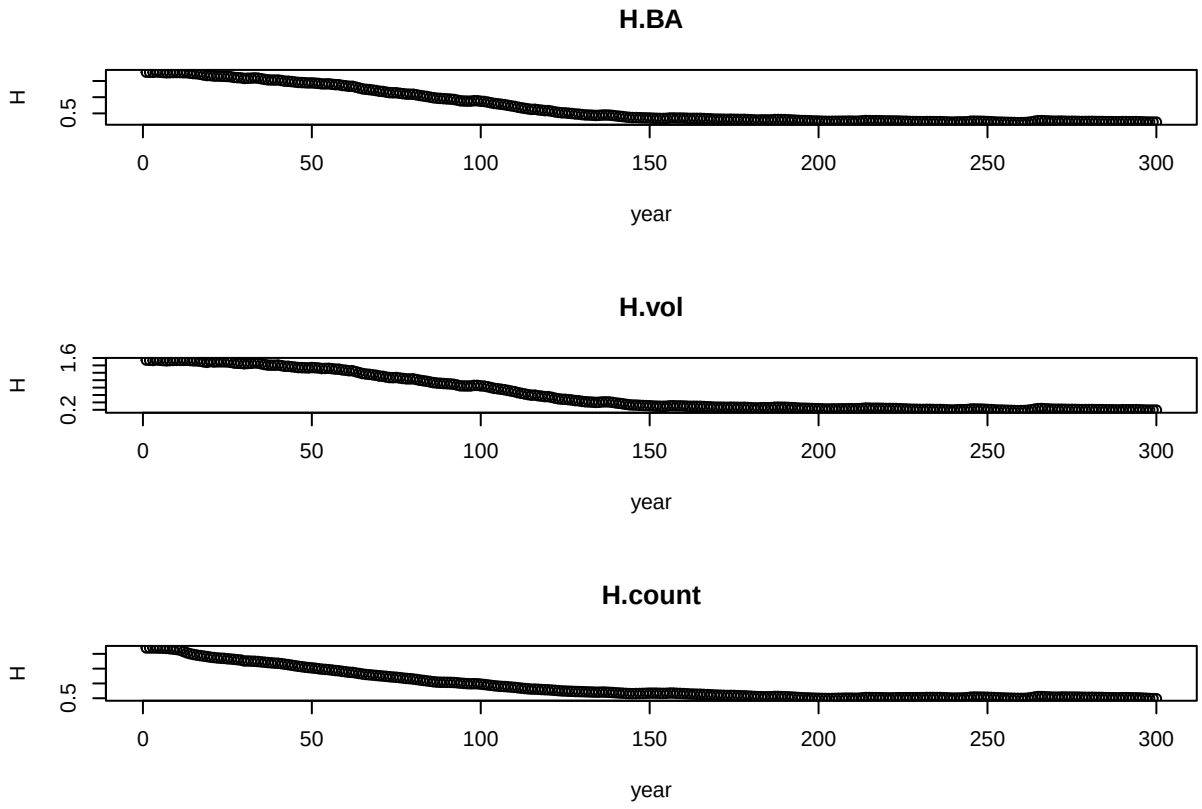
```
## [1] "C:/iLand/2023/browsing_revision/db_test/B2_refclim_w3.sqlite"
##  [1] "abeUnit"           "barkbeetle"        "carbon"
##  [4] "carbonflow"        "dynamicstand"      "landscape"
##  [7] "landscape_removed" "runinfo"           "soilinput"
## [10] "wind"
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```



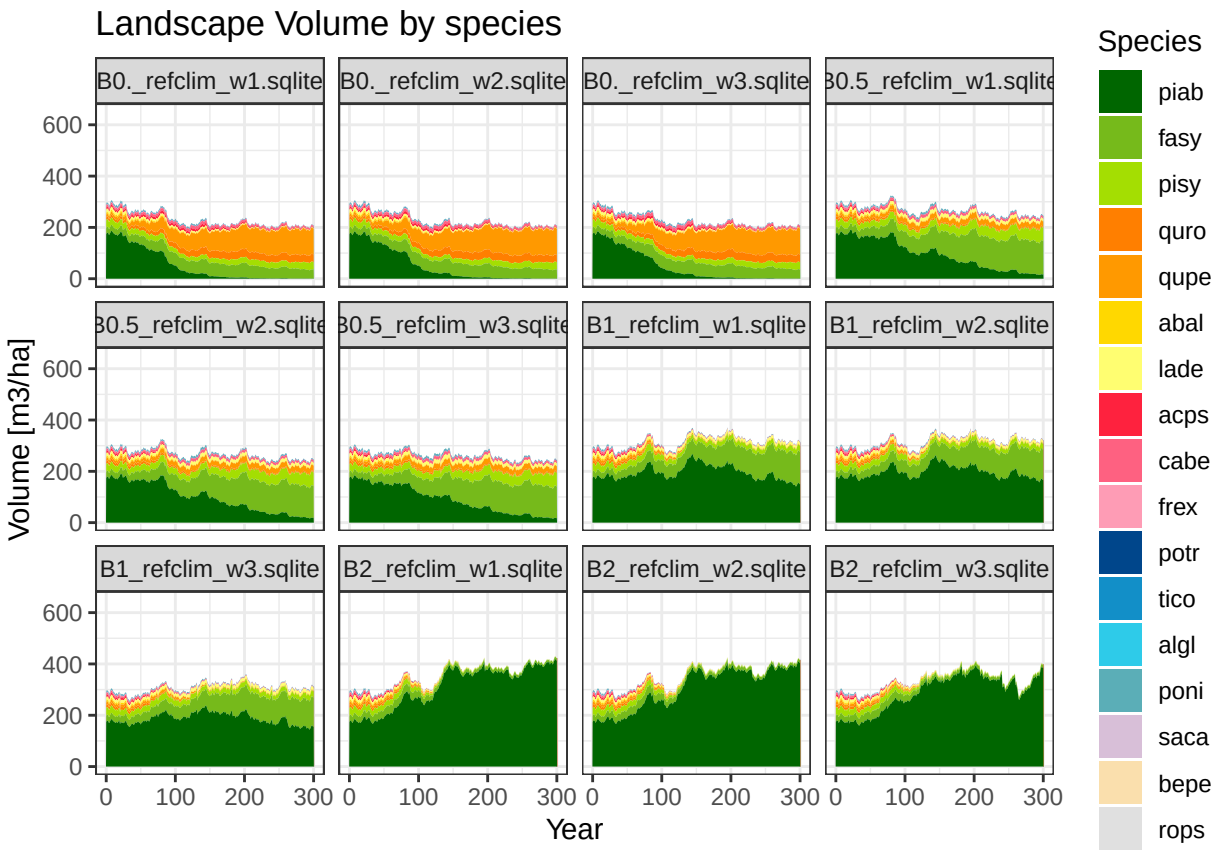

```
## # A tibble: 6 x 4
##    year VOL.tot BA.tot count.tot
##   <int>   <dbl>  <dbl>     <dbl>
## 1     1    299.   30.6     1000.
## 2     2    296.   30.4     1004.
## 3     3    294.   30.3     1007.
## 4     4    284.   29.3      987.
## 5     5    288.   29.8      996.
## 6     6    297.   30.8     1002.
```

```
## # A tibble: 6 x 5
## # Groups:   year [1]
##    year species    VOL     BA  count
##   <int> <chr>    <dbl>  <dbl>  <dbl>
## 1     1 abal     6.59  0.555   17.2
## 2     1 acps     7.89  1.05    67.7
## 3     1 algl     0.604 0.0859   3.57
## 4     1 bepe     0.401 0.0946   5.30
## 5     1 cabe     7.91  1.22    65.7
## 6     1 fasy    23.7   2.47   121.
##   [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [26] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [51] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
##  [76] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 16 16 16 16 16 16 16 17 17 17
## [101] 17 17 17 17 17 17 16 16 16 16 16 16 16 16 15 15 15 15 15 15 15 15 15 15 15
## [126] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 14 14 14 14 14 14
## [151] 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
## [176] 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
## [201] 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 13 13 13 13 13 13 13
## [226] 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
## [251] 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 11 11 11 10 10 10 10 10 10
## [276] 10 10 10 10 10 10 10 10 10 10 10  9  9  9  9  9  9  9  9  9  9  9  9  9  9
## plot window will lay out plots in a 3 by 1 matrix
```



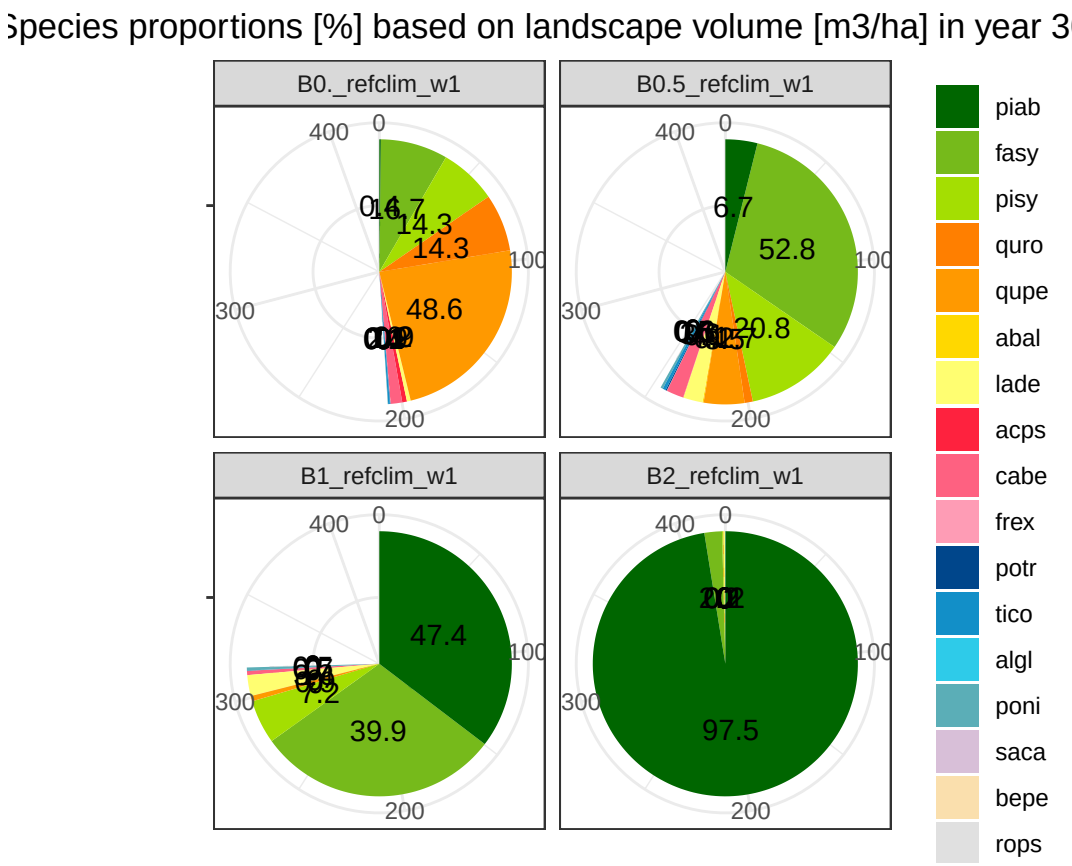
```
# This tells the colors:

species.we.have<-unique(lnd$species)                                    # IT IS SAYING WHICH SP


# LIST OF ALL POSSIBLE SPECIES

cols.all=c( "rops"="#e0e0e0", "acpl"="#A9A9A9",   "alin"="#696969", "alvi"="#2e2e2e",
            "bepe"="#fadfad",
            "casa"="#7eeadf", "coav"="#20c6b6",
            "tipl"="#645394", "ulgl"="#311432" ,
            "saca"="#D8BFD8",  "soar"="#DDA0DD", "soau"="#BA55D3",
            "pice"="#D27D2D", "pini"="#a81c07",
            "algl"="#2ECBE9","tico"="#128FC8",  "potr"="#00468B","poni"="#5BAEB7",
            "frex"="#fe9cb5","cabe"="#fe6181","acps"="#fe223e",
            "lade"="#FFFE71","abal"="#FFD800", "pisy"="#A4DE02",
            "fasy"="#76BA1B", "piab"="#006600",
            "quro"="#FF7F00", "qupe"="#FF9900", "qupu"="#CC9900"
)


# COLORATION ORDER FOR ALL THE POSSIBLE SPECIES

new_order_gg.all=c("alvi","alin", "acpl", "rops","bepe" ,"coav", "casa", "ulgl", "tipl",  "soau", "soar
                   "poni", "algl", "tico", "potr",  "frex","cabe", "acps",  "lade", "abal",  "qupu", "q


# This will show at the end only the species we really have on the landscape.

cols<-cols.all[names(cols.all) %in% species.we.have]
new_order_gg<- new_order_gg.all[new_order_gg.all %in% species.we.have]

# STARTING PLOTS


# Make a plot with ggplot, volume, colored by species for the transitional period for Clear cut managem
```

```
# PLOT LANDSCAPE VOLUME PLOT FOR CASES (GEOM AREA)

ggplot(lnd, aes(year,volume_m3, fill=factor(species, levels=new_order_gg)))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Landscape Volume by species")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="Volume [m3/ha]",fill = "Species")+
  theme(plot.title = element_text(hjust = 0.5))+
  ylim(0,650)+
  theme_bw()
```



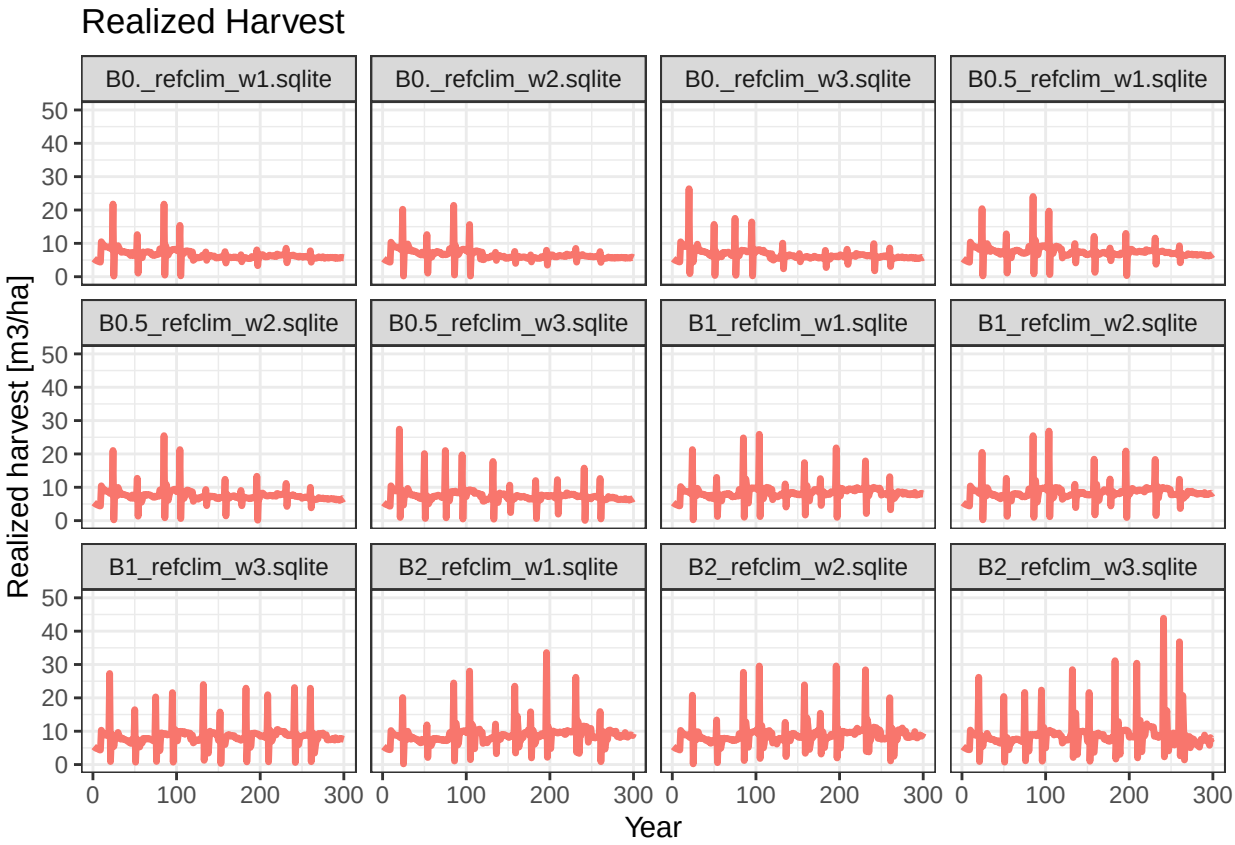

```
# Pie chart year 0

# Filter per the year and create the percentage data frame

B0._refclim_w1 <- lnd %>% filter(run=="B0._refclim_w1.sqlite" & year==300)
B0._refclim_w1_per<-B0._refclim_w1 %>% mutate( sumvol=sum(volume_m3)) %>% mutate(perc.vol=100*volume_m3/

B0.5_refclim_w1 <- lnd %>% filter(run=="B0.5_refclim_w1.sqlite" & year==300)
B0.5_refclim_w1_per<-B0.5_refclim_w1 %>% mutate( sumvol=sum(volume_m3)) %>% mutate(perc.vol=100*volume_

B1_refclim_w1 <- lnd %>% filter(run=="B1_refclim_w1.sqlite" & year==300)
B1_refclim_w1_per<-B1_refclim_w1 %>% mutate( sumvol=sum(volume_m3)) %>% mutate(perc.vol=100*volume_m3/su

B2_refclim_w1 <- lnd %>% filter(run=="B2_refclim_w1.sqlite" & year==300)
B2_refclim_w1_per<-B2_refclim_w1 %>% mutate( sumvol=sum(volume_m3)) %>% mutate(perc.vol=100*volume_m3/su


r1wb<-rbind(B0._refclim_w1_per,B0.5_refclim_w1_per, B1_refclim_w1_per, B2_refclim_w1_per)

x7wb <- ggplot(r1wb, aes(x="", y=volume_m3, fill=factor(species,levels=new_order_gg))) +
  geom_bar(stat="identity", width=1, show.legend = T) +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  facet_wrap(~case, ncol=2)+
  coord_polar("y", start=0) +
  geom_text(aes(label = paste0( round(perc.vol, 1)  )),  position = position_stack(vjust=0.5)) +
  labs(x = NULL, y = NULL, fill = NULL)+
```

```
  ggtitle("Species proportions [%] based on landscape volume [m3/ha] in year 300")+
  theme_bw()
x7wb + theme(plot.title = element_text(hjust = 0.5))
```




```
# (SHOULD BE REALIZED) PLOT 2 "Y" AXIS WITH relationship between realized harvest and volume increasing
# Total realized harvest at landscape level in average per ha

ggplot(aUnit, aes(year,realizedHarvest, color=case))+
  geom_line(size=1.2, show.legend = F)+
  facet_wrap(~run, ncol=4)+
  ylim(0,50)+
  ggtitle("Realized Harvest")+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("Realized harvest [m3/ha]")+
  xlab("Year")+
  theme_bw()
```

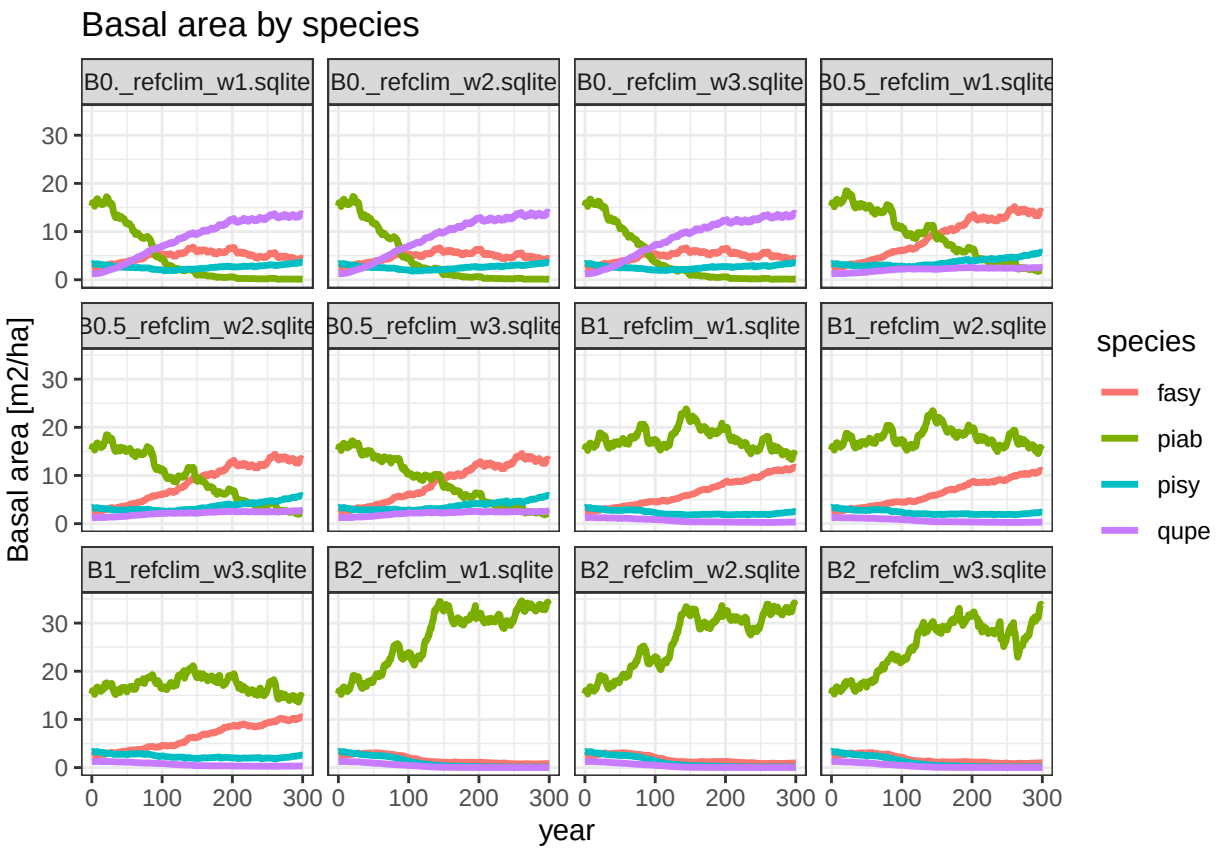
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



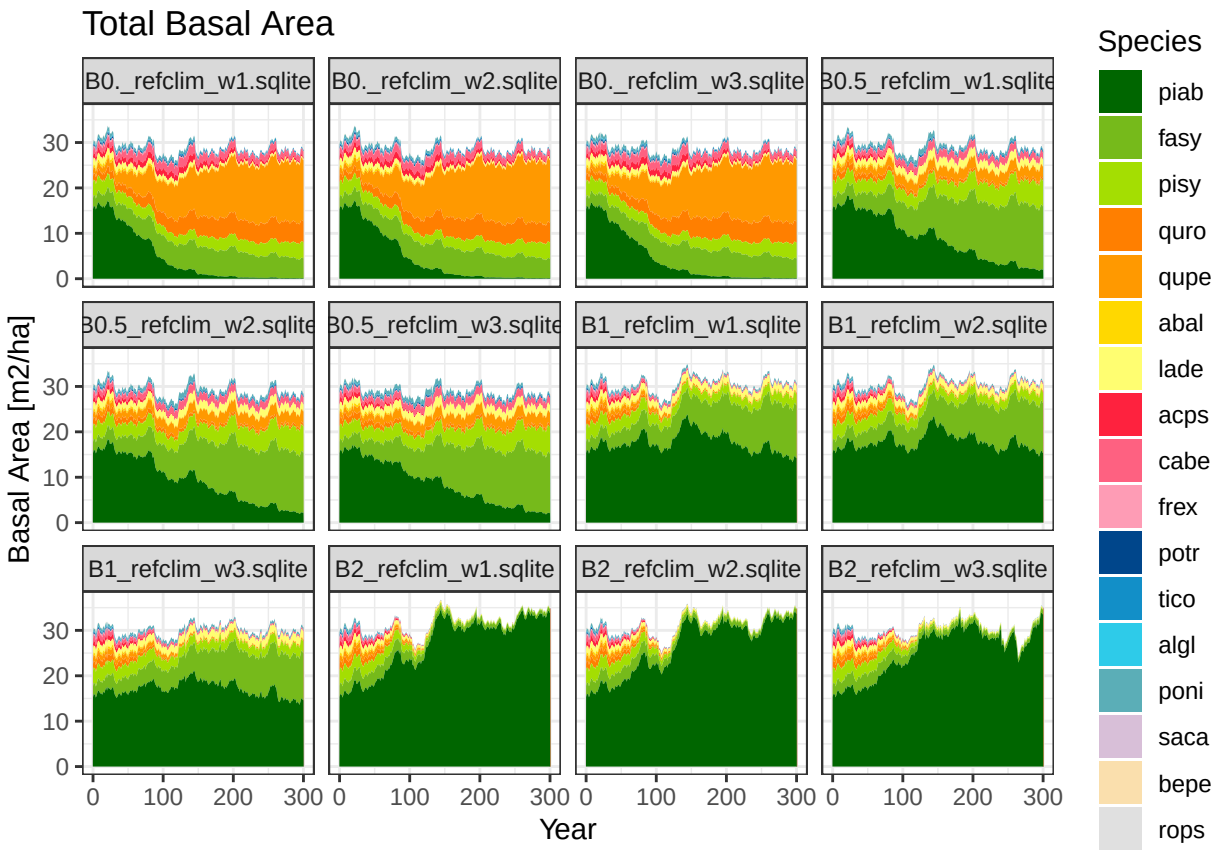

```
# SPECIES specificaly BA:

#species.to.keep<-c("piab","pisy", "fasy","qupe")

#lnd2 <- lnd %>% filter(species %in% species.to.keep)

#ggplot(data=lnd2, aes(x=year, y=basal_area_m2, colour=species)) +
#  geom_line(size=1.2)+
#  scale_colour_manual(values = c("#76BA1B","#006600", "#A4DE02", "orange"))+
#  ggtitle("Clearcut management in brow pressure 0") +
#  theme(plot.title = element_text(hjust = 0.5))+
#  ylab("Basal area [m2/ha]")+
#  theme_bw()
```

```
# PLOT BASAL AREA GEOM_LINE AT LANDSCAPE LEVEL BY SPECIES SELECTED

# SPECIES specifically BA:

species.to.keep<-c("piab", "fasy","qupe", "pisy")


lnd2 <- lnd %>% filter(species %in% species.to.keep)

ggplot(data=lnd2, aes(x=year, y=basal_area_m2, color=species)) +
  geom_line(size=1.2)+
  ggtitle("Basal area by species") +
```

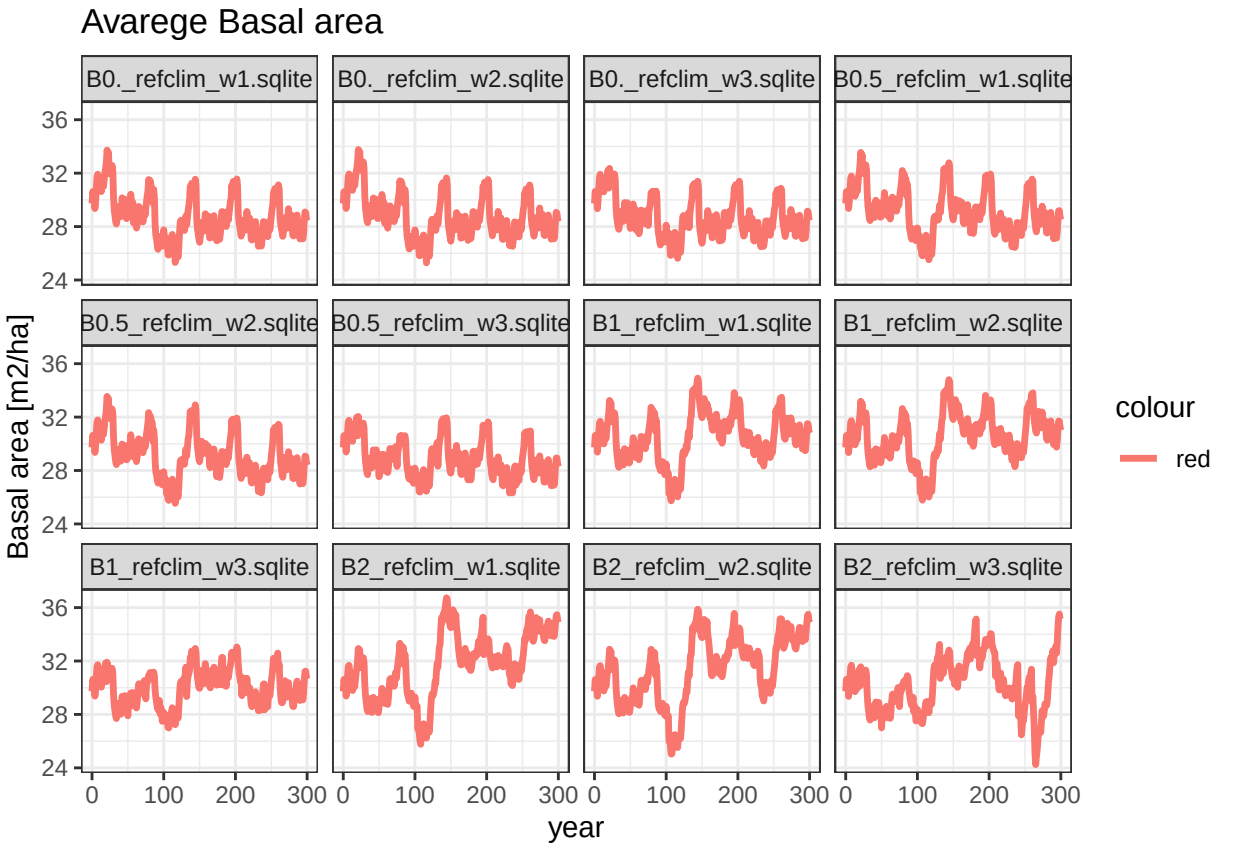
```
  facet_wrap(~run, ncol=4)+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("Basal area [m2/ha]")+
  theme_bw()
```



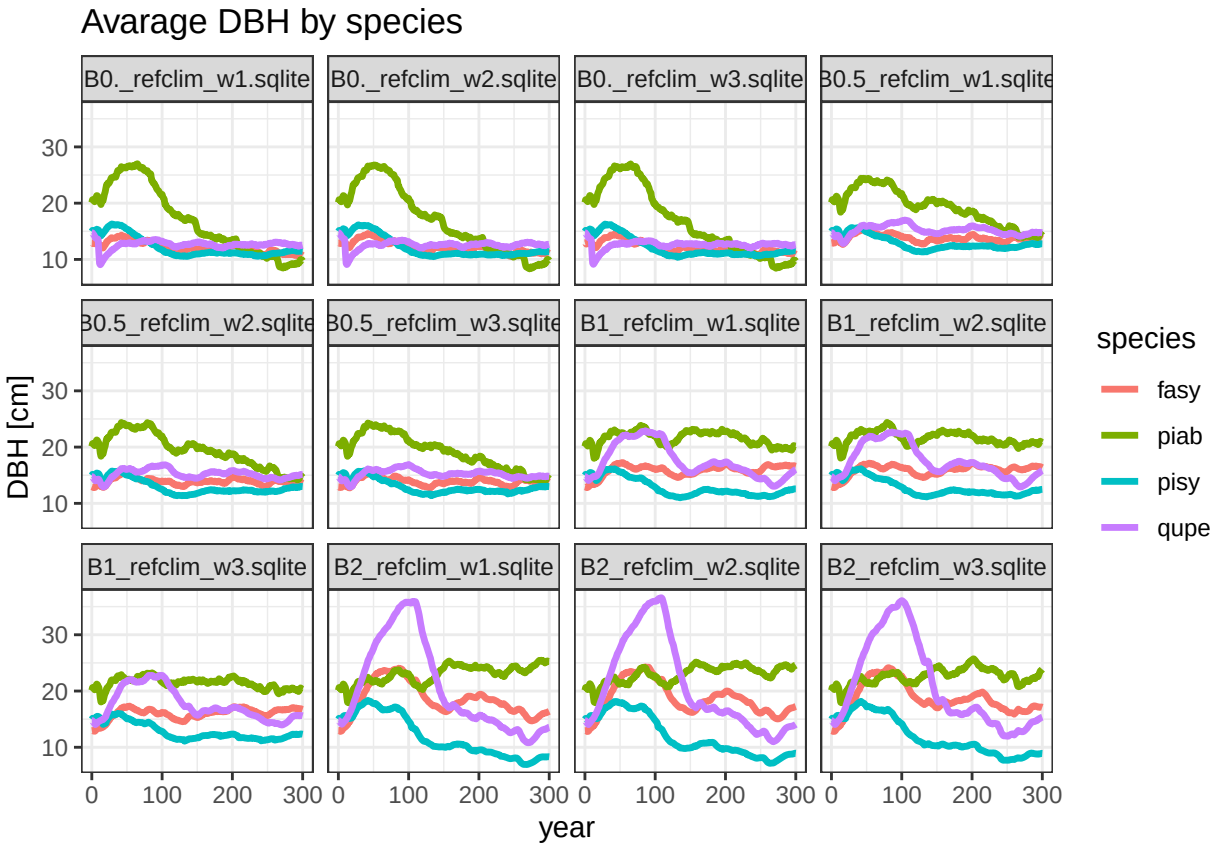

```
# PLOT TOTAL AVG BASAL AREA AT LANDSCAPE LEVEL BY SPECIES

ggplot(lnd, aes(year, basal_area_m2, fill=factor(species, levels=new_order_gg)))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Total Basal Area")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="Basal Area [m2/ha]",fill = "Species")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



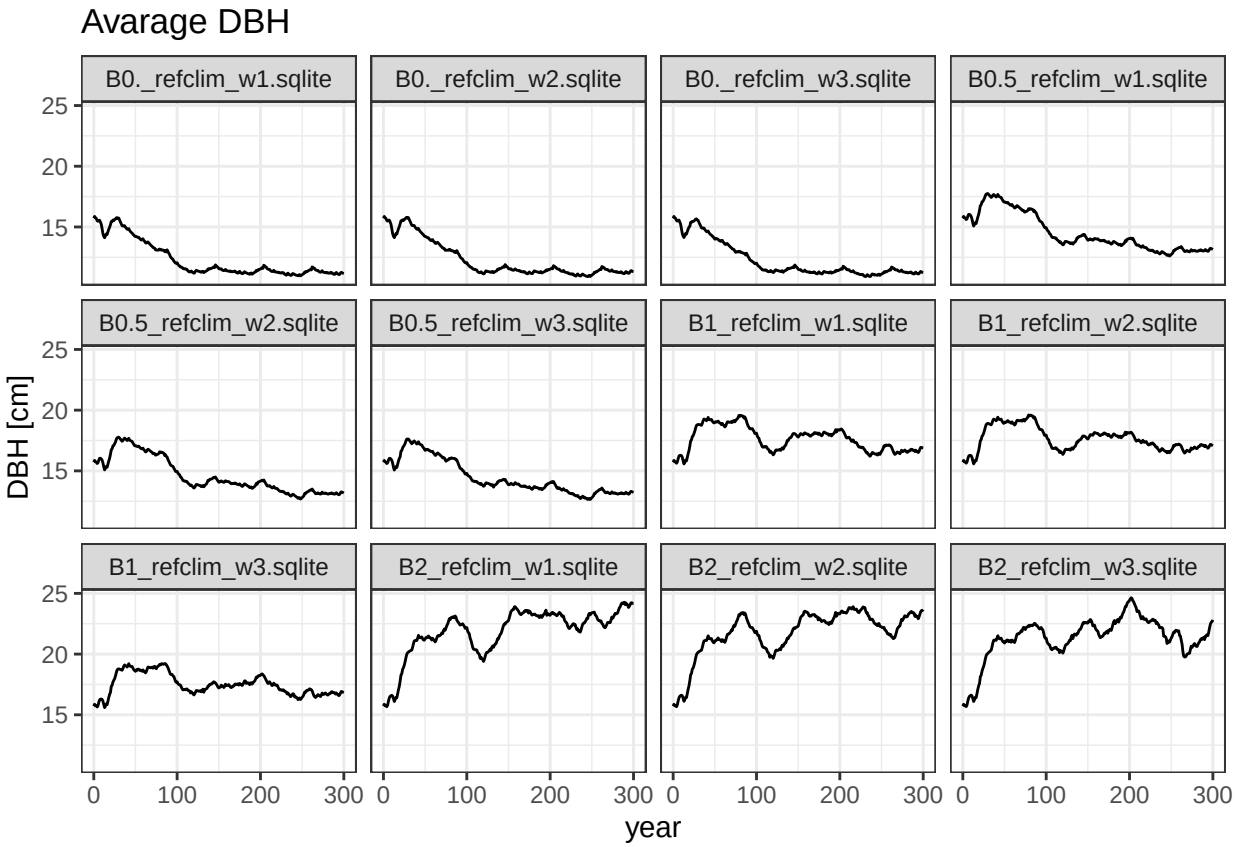

```
# PLOT SUM BASAL AREA AT LANDSCAPE

ggplot(data=dys, aes(x=year, y=basalarea_sum/landscape.area, color="red")) +
  geom_line(size=1.2)+
  ggtitle("Avarege Basal area") +
  facet_wrap(~run, ncol=4)+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("Basal area [m2/ha]")+
  theme_bw()
```




```
# PLOT DBH GEOM_LINE AT LANDSCAPE LEVEL BY SPECIES

species.to.keep<-c("piab", "fasy","qupe", "pisy")


lnd2 <- lnd %>% filter(species %in% species.to.keep)

ggplot(data=lnd2, aes(x=year, y=dbh_avg_cm, color=species)) +
  geom_line(size=1.2)+
  ggtitle("Avarage DBH by species") +
  facet_wrap(~run, ncol=4)+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("DBH [cm]")+
  theme_bw()
```

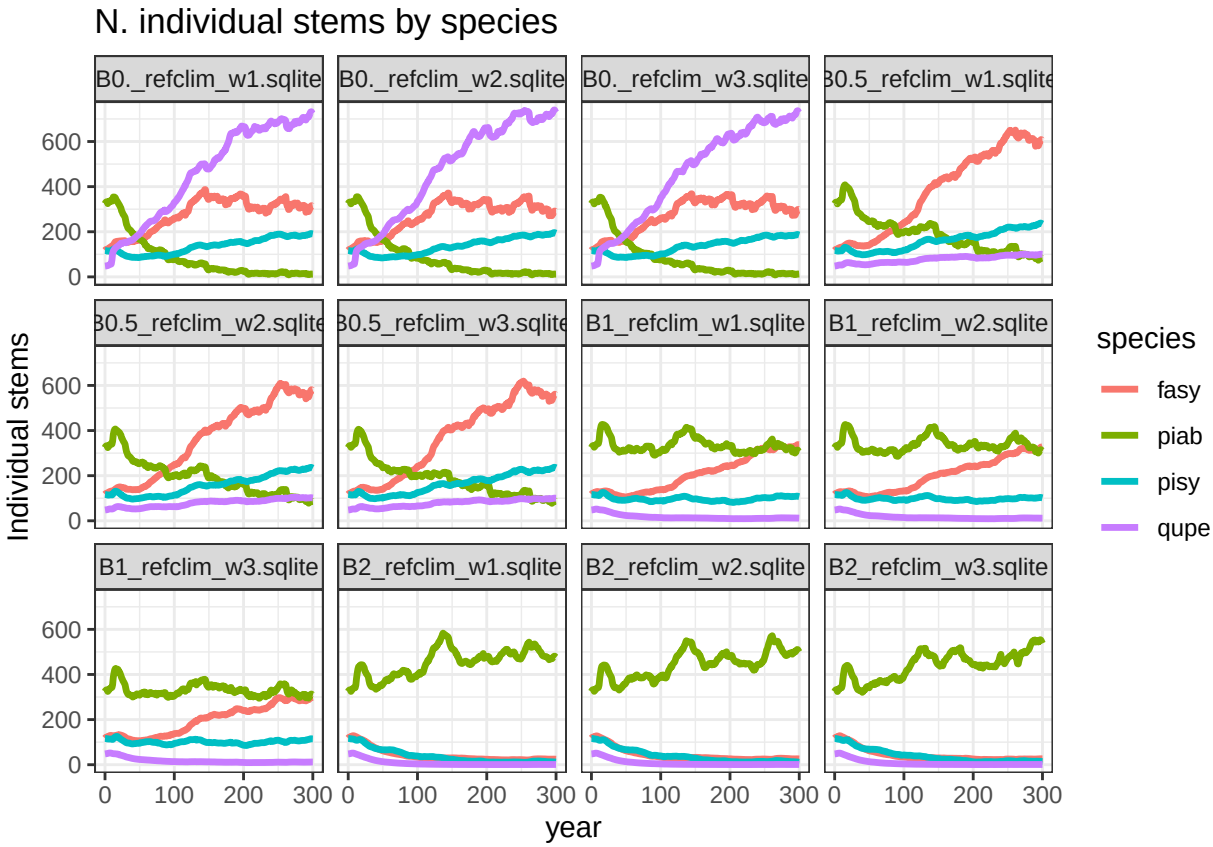



```
# PLOT DBH GEOM_AREA AT LANDSCAPE LEVEL AVARAGE ALL SP TOGETHER

ggplot(data=dys, aes(x=year, y=dbh_mean)) +
  geom_line()+
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Avarage DBH") +
  facet_wrap(~run, ncol=4)+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("DBH [cm]")+
  theme_bw()
```

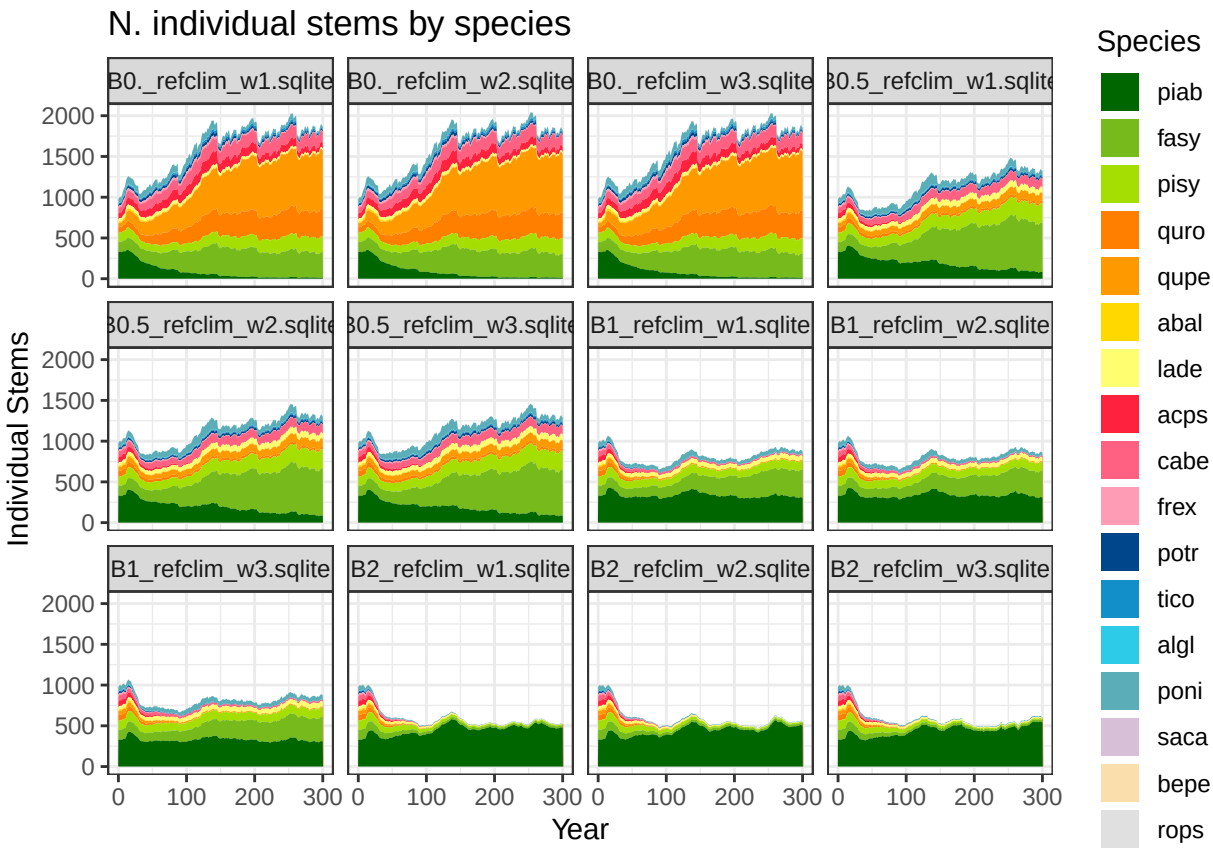



```
# SD DBH / try with violin plots

# ggplot(data=dys, aes(x=year, y=dbh_sd)) +
#  geom_area(size=1.2)+
#  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
#  ggtitle("Avarage DBH by species") +
#  facet_wrap(~run, ncol=1)+
#  theme(plot.title = element_text(hjust = 0.5))+
#  ylab("DBH [cm]")+
#  theme_bw()
```

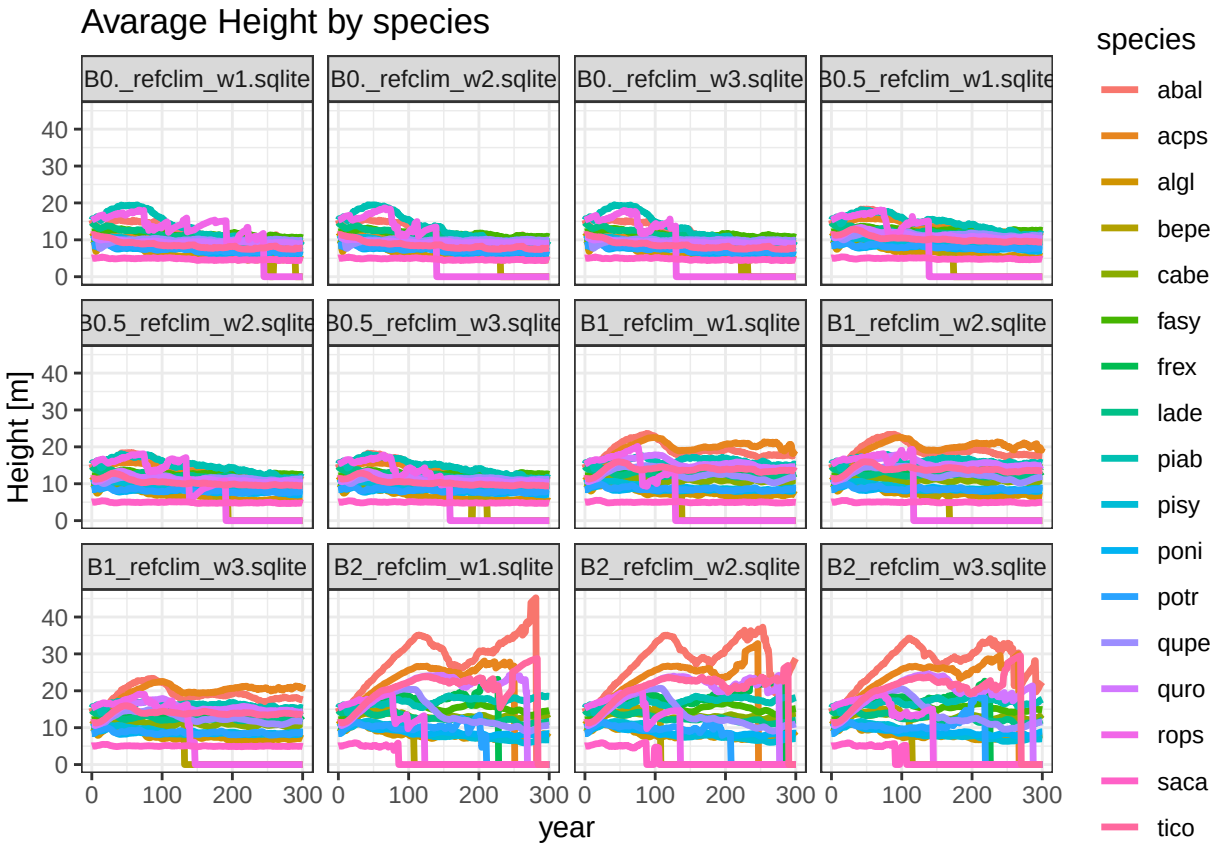
```
# PLOT NUMBER OF STEMS GEOM_LINE AT LANDSCAPE LEVEL BY SPECIES

# lnd2 <- lnd %>% filter(species %in% species.to.keep)    --- ADD IF YOU WANT SELECT SPECIES TO KEEP

ggplot(data=lnd2, aes(x=year, y=count_ha, color=species)) +
  geom_line(size=1.2)+
  ggtitle("N. individual stems by species") +
  facet_wrap(~run, ncol=4)+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("Individual stems")+
  theme_bw()
```



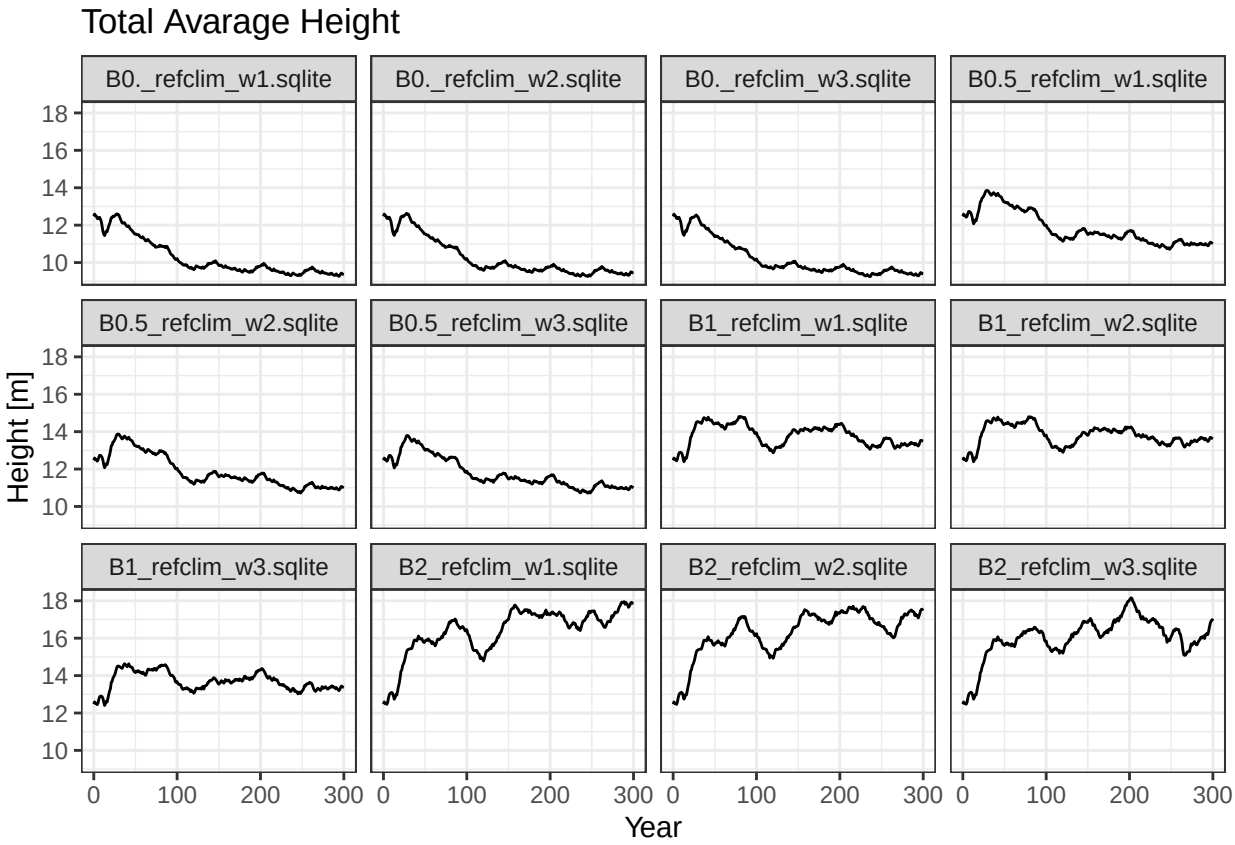

```
# PLOT NUMBER OF STEMS GEOM_AREA AT LANDSCAPE LEVEL BY SPECIES

ggplot(lnd, aes(x=year, y=count_ha, fill=factor(species, levels=new_order_gg)))+
  geom_area(size=1.2)+
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("N. individual stems by species") +
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="Individual Stems", fill = "Species")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



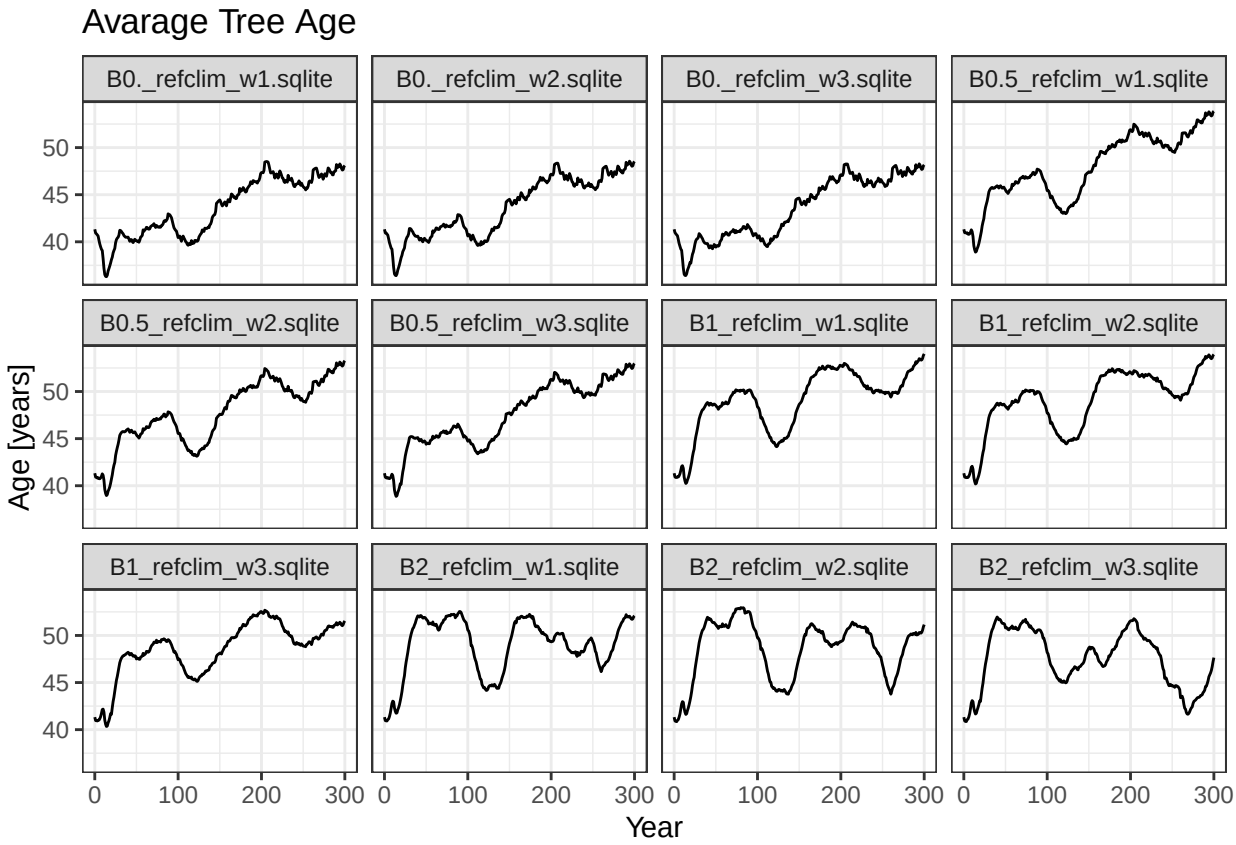

```
# PLOT HEIGHT GEOM_LINE AT LANDSCAPE LEVEL BY SPECIES

# lnd2 <- lnd %>% filter(species %in% species.to.keep)    #---   ADD IF YOU WANT SELECT SPECIES TO KEEP

ggplot(data=lnd, aes(x=year, y=height_avg_m, color=species)) +
  geom_line(size=1.2)+
  ggtitle("Avarage Height by species") +
  facet_wrap(~run, ncol=4)+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("Height [m]")+
  theme_bw()
```



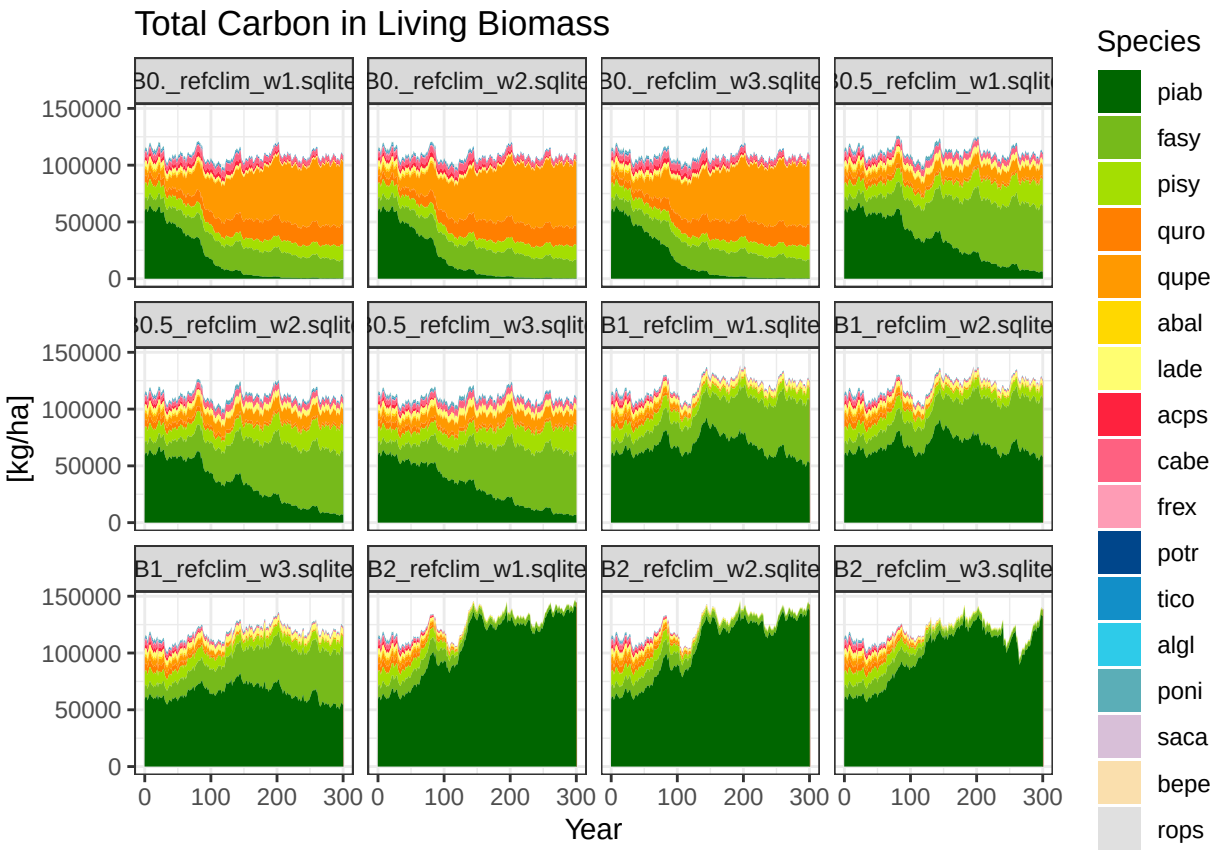

```
# PLOT TOTAL HEIGHT GEOM_AREA AT LANDSCAPE LEVEL BY SPECIES

ggplot(dys, aes(x=year, y=height_mean))+
  geom_line() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Total Avarage Height")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="Height [m]")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



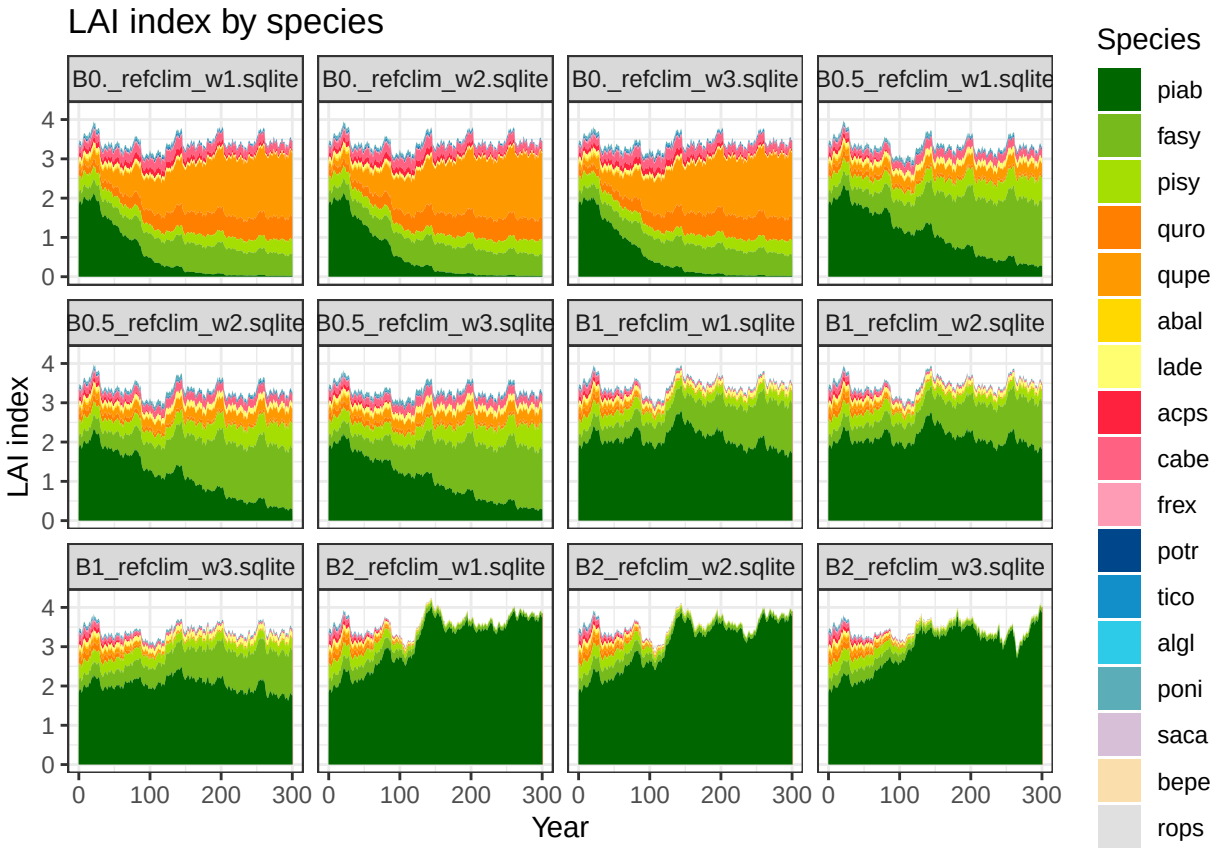

```
# AGE

ggplot(dys, aes(x=year, y=age_mean))+
  geom_line() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Avarage Tree Age")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="Age [years]")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



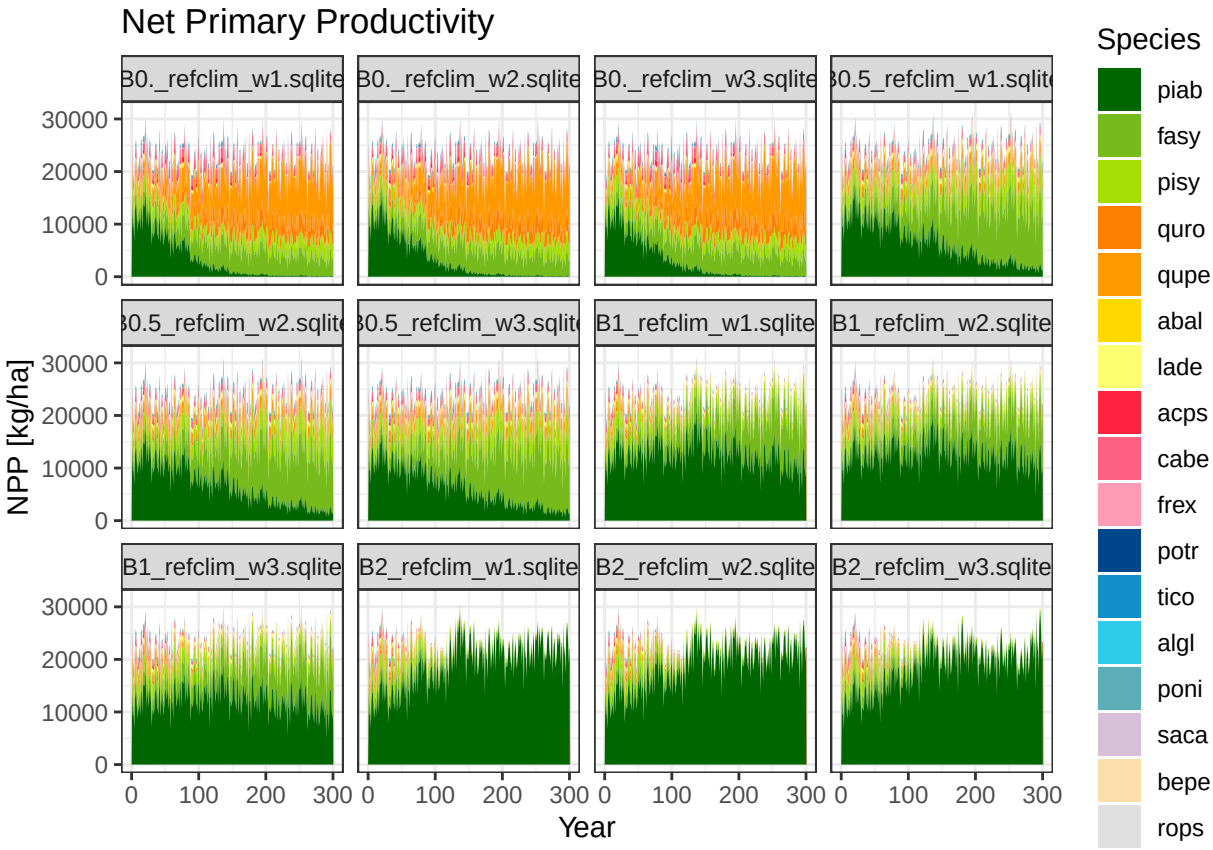

```
# Total Carbon in Kg (total_carbon_kg   double  total carbon in living biomass (aboveground compartment

ggplot(lnd, aes(year, total_carbon_kg, fill=factor(species, levels=new_order_gg)))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Total Carbon in Living Biomass")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="[kg/ha]",fill = "Species")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



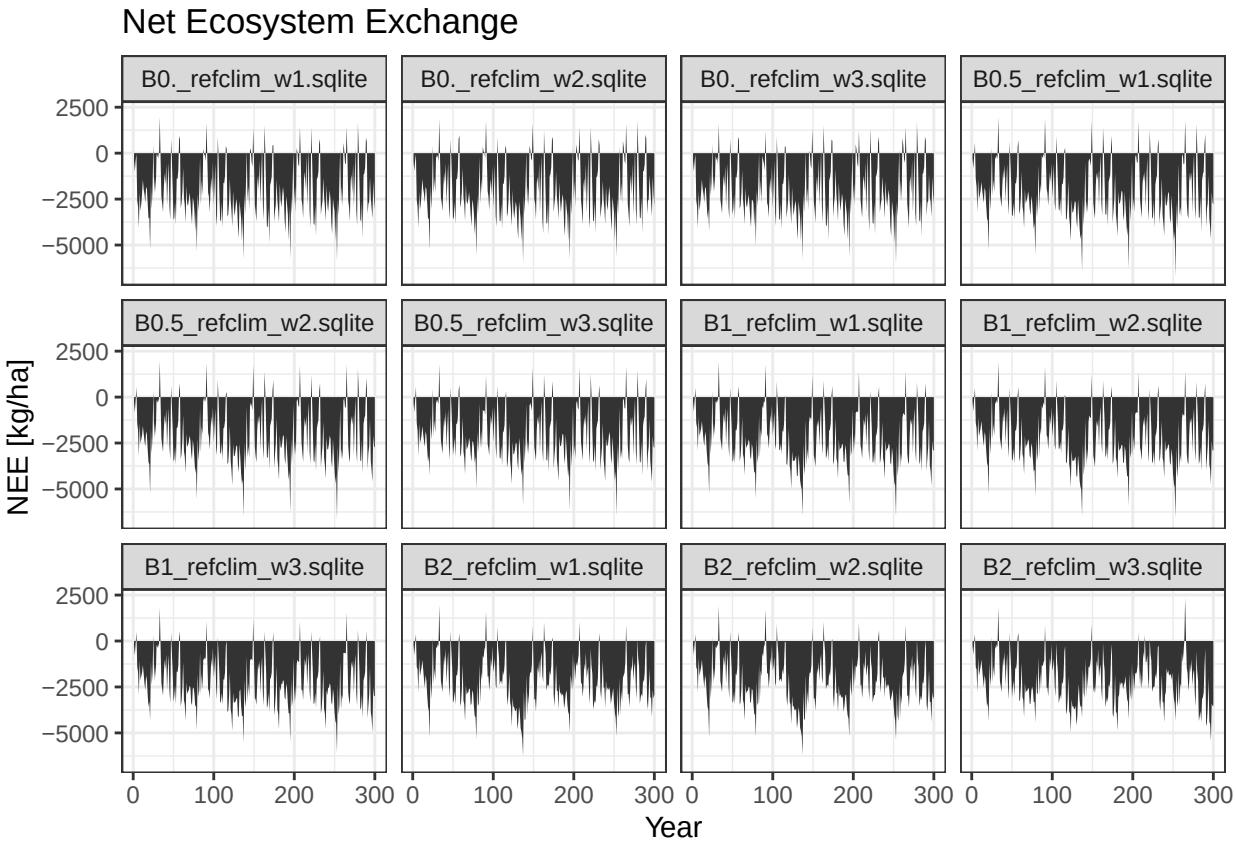

```
# PLOT LAI AT LANDSCAPE LEVEL BY SPECIES

ggplot(lnd, aes(year, LAI, fill=factor(species, levels=new_order_gg)))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("LAI index by species")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="LAI index",fill = "Species")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



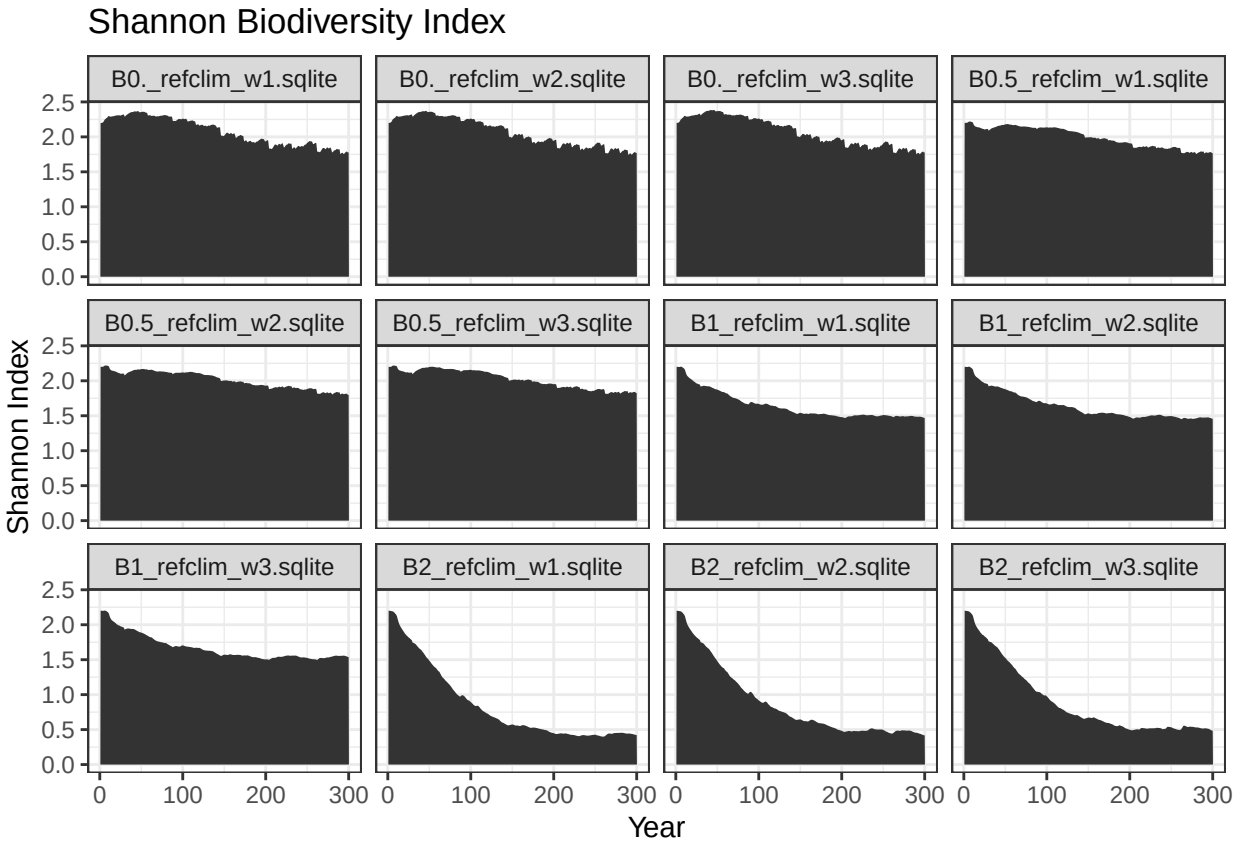

```
# PLOT NPP AT LANDSCAPE LEVEL BY SPECIES

ggplot(lnd, aes(year, NPP_kg, fill=factor(species, levels=new_order_gg)))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Net Primary Productivity")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="NPP [kg/ha]",fill = "Species")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



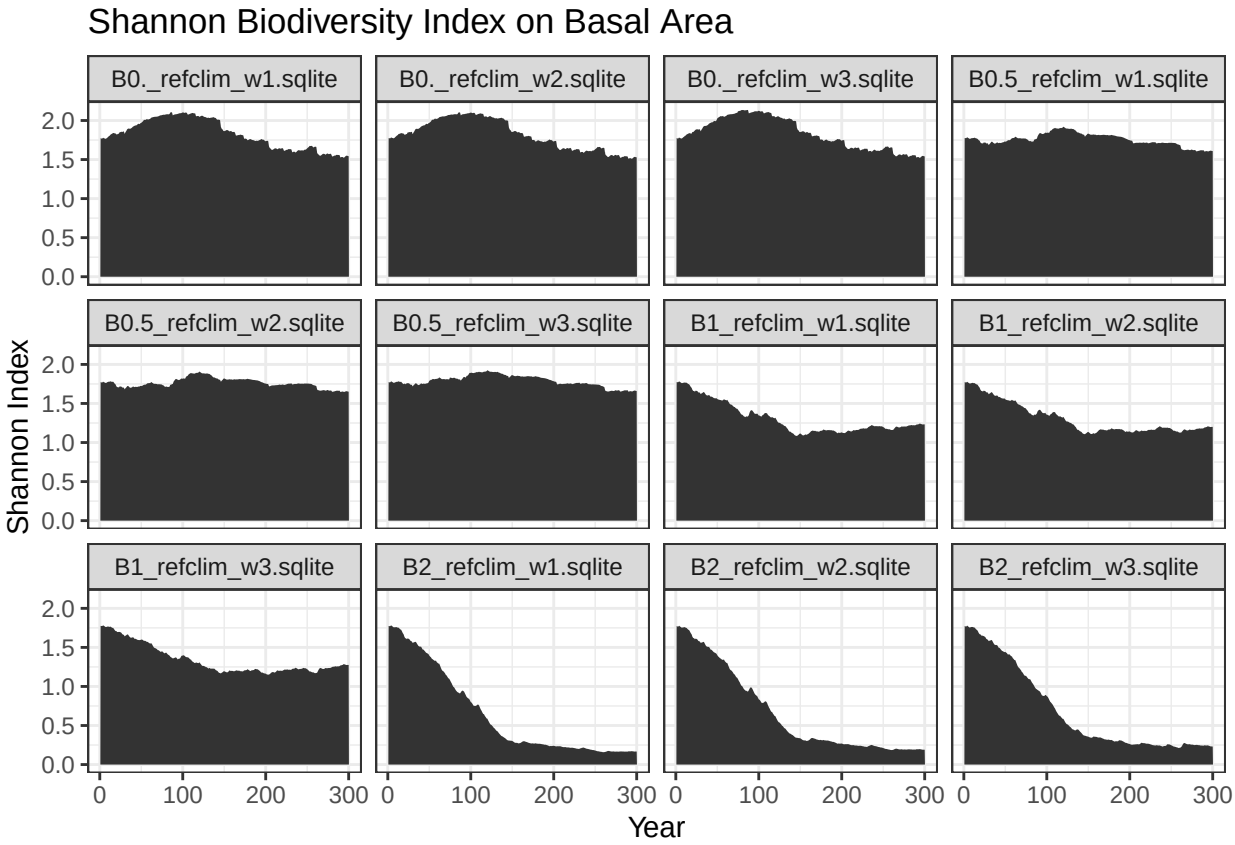

```
# PLOT NEE AT LANDSCAPE LEVEL

ggplot(variables.all, aes(year, NEE))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Net Ecosystem Exchange")+
  facet_wrap(~case, ncol=4)+
  labs(x = "Year",y="NEE [kg/ha]")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



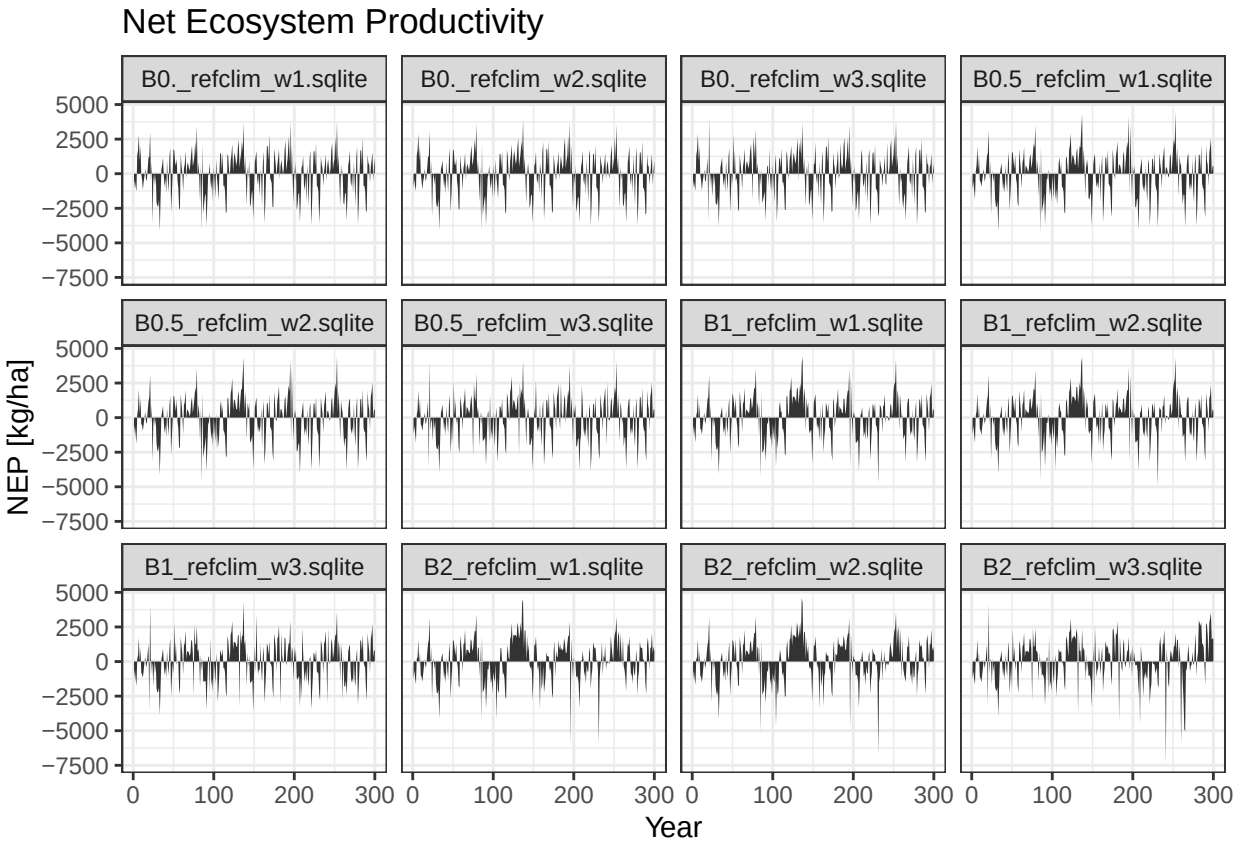

```
# PLOT H count AT LANDSCAPE LEVEL

ggplot(variables.all, aes(year, H.count))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Shannon Biodiversity Index")+
  facet_wrap(~case, ncol=4)+
  labs(x = "Year",y="Shannon Index")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



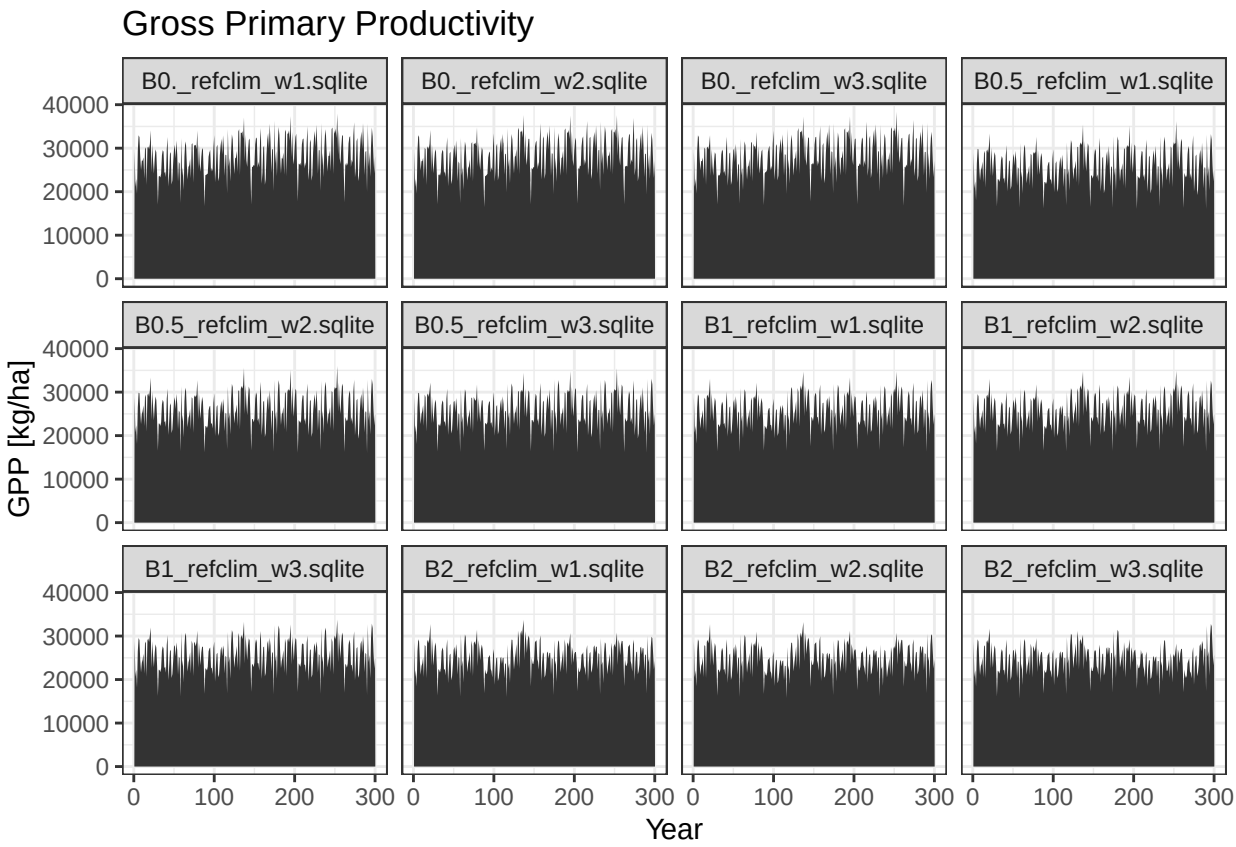

```
# PLOT H basal area AT LANDSCAPE LEVEL

ggplot(variables.all, aes(year, H.BA))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Shannon Biodiversity Index on Basal Area")+
  facet_wrap(~case, ncol=4)+
  labs(x = "Year",y="Shannon Index")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```




```
# PLOT H basal area AT LANDSCAPE LEVEL

ggplot(variables.all, aes(year, NEP))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Net Ecosystem Productivity")+
  facet_wrap(~case, ncol=4)+
  labs(x = "Year",y="NEP [kg/ha]")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```

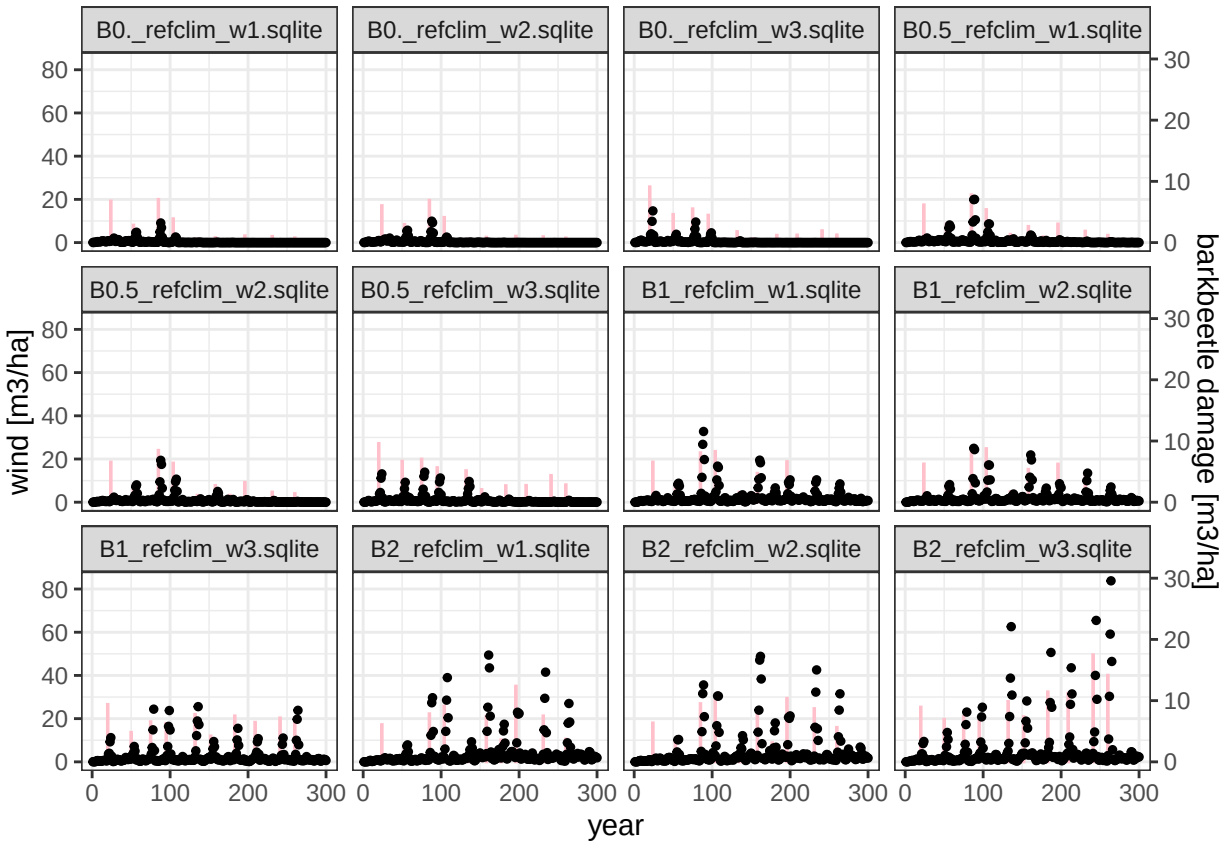



```
# PLOT GPP AT LANDSCAPE LEVEL

ggplot(variables.all, aes(year, GPP))+
  geom_area() +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Gross Primary Productivity")+
  facet_wrap(~case, ncol=4)+
  labs(x = "Year",y="GPP [kg/ha]")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()
```



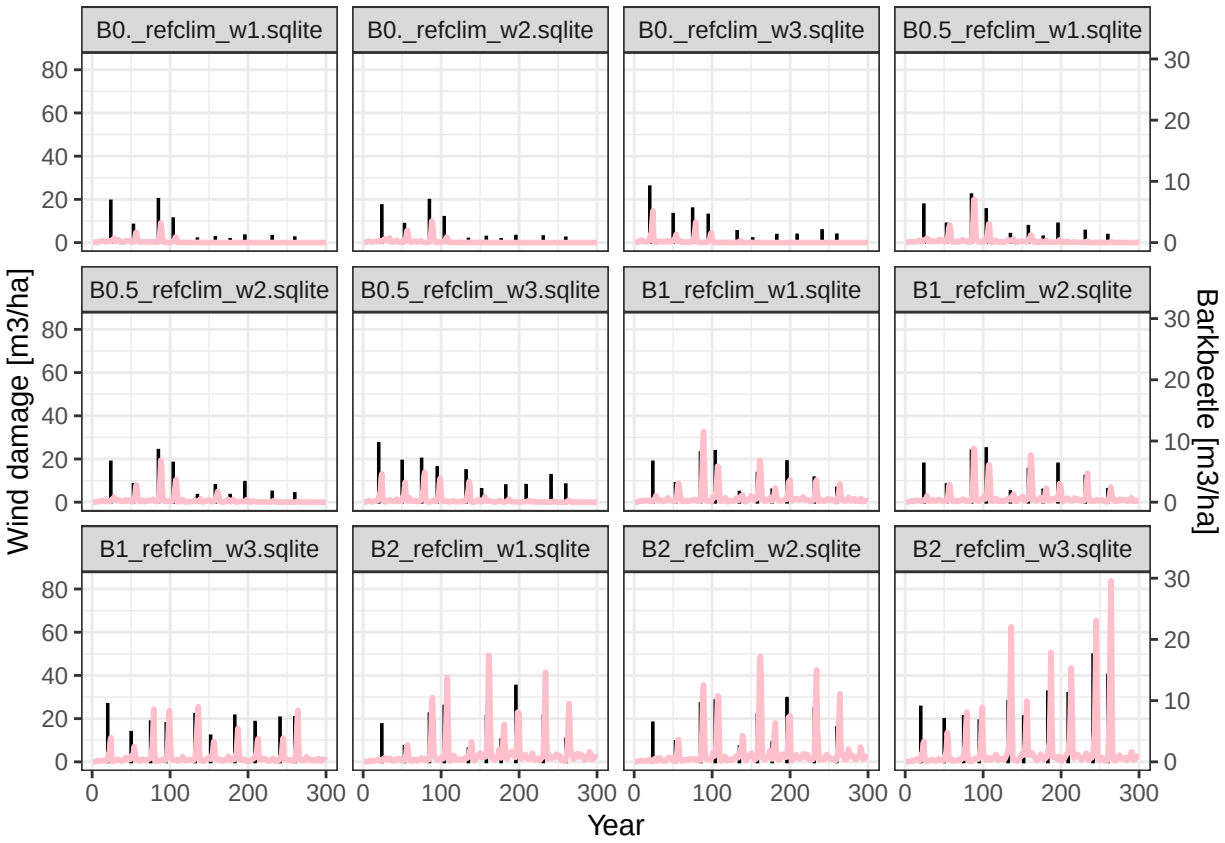

```
# PLOT SECOND Y AXIS FOR KILLED VOLUME BY DISTURBANCES IN LANDSCAPE AVG LINE 75

area<-lnd$area[1]
ylim.bb <- c(0, 600000)
ylim.w <- c(0, 1700000)

b <- diff(ylim.w)/diff(ylim.bb)
a <- ylim.w[1] - b*ylim.bb[1]

# TO MAKE 2 LINE AND 2 DIFFERENT SCALE PLOT "https://stackoverflow.com/questions/3099219/ggplot-with-2-
ggplot(damage.all,aes(year,killedVolume/area))+
  geom_col(fill="pink",col="pink")+
  geom_point(aes(y = a+ barkbeetle/area*b), data = damage.all,size=1) +
  scale_y_continuous(name="wind [m3/ha]", sec.axis = sec_axis(~ (. - a)/b,name = "barkbeetle damage [m3
  facet_wrap(~case, ncol=4)+
  theme_bw()
```

```
## Warning: Removed 3480 rows containing missing values (‘position_stack()‘).
```




```
# variant 2

b <- diff(ylim.w)/diff(ylim.bb)
a <- ylim.w[1] - b*ylim.bb[1]

# TO MAKE 2 LINE AND 2 DIFFERENT SCALE PLOT "https://stackoverflow.com/questions/3099219/ggplot-with-2-
ggplot(damage.all,aes(year,killedVolume/area))+
  geom_col(fill="grey",col="black", size=0.5)+
  geom_line(aes(y = a+ barkbeetle/area*b), data = damage.all, size=1, col="pink") +
  scale_y_continuous(name="Wind damage [m3/ha]", sec.axis = sec_axis(~ (. - a)/b,name = "Barkbeetle [m3/
  xlab("Year")+
  facet_wrap(~case, ncol=4)+
  theme_bw()
```

```
## Warning: Removed 3480 rows containing missing values (‘position_stack()‘).
```



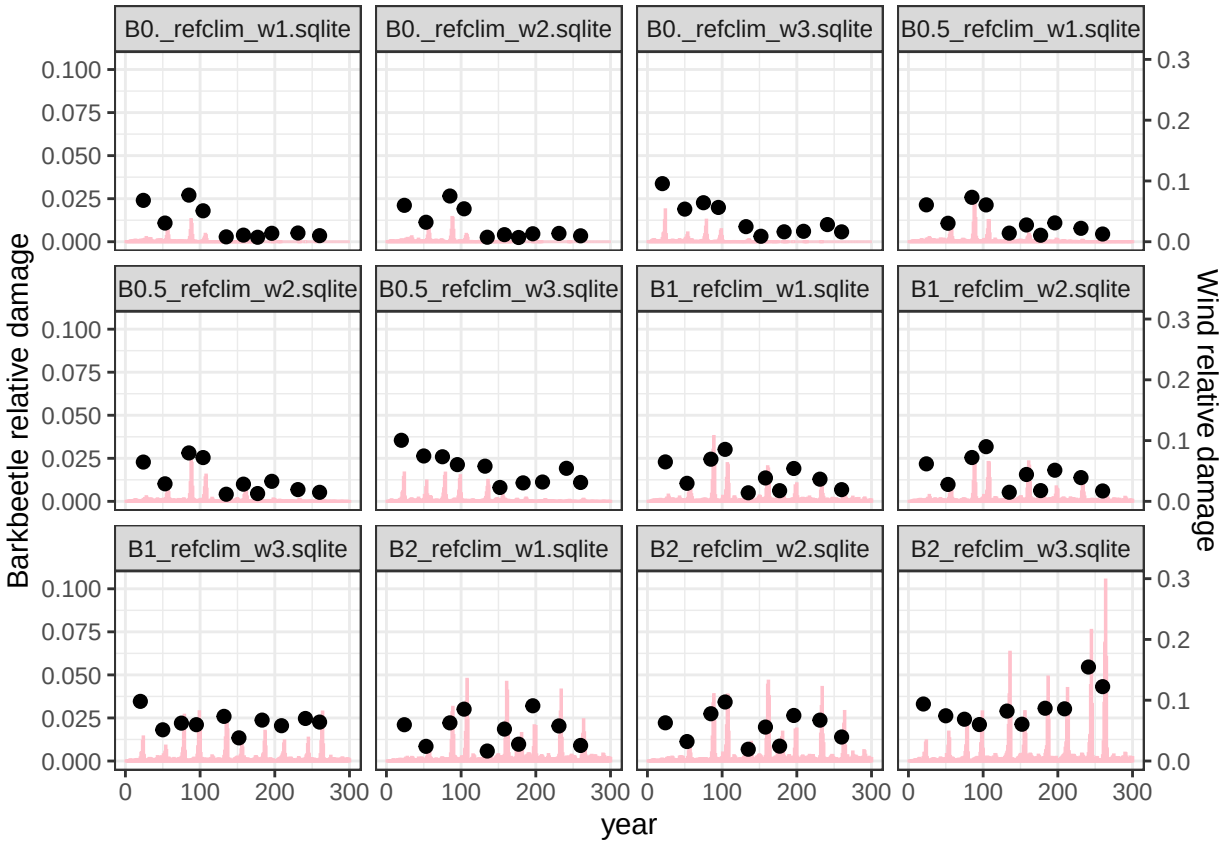

```
# Here gave me the right values
absolute_damage <- (damage.all %>% group_by(case) %>% summarise(mean(barkbeetle/area),mean(na.omit(kill
absolute_damage
```

```
## # A tibble: 12 x 3
##    case                    `mean(barkbeetle/area)` mean(na.omit(killedVolume/ar~1
##    <chr>                                     <dbl>                          <dbl>
##  1 B0.5_refclim_w1.sqlite                    0.289                           9.50
##  2 B0.5_refclim_w2.sqlite                    0.290                          10.1
##  3 B0.5_refclim_w3.sqlite                    0.307                          13.9
##  4 B0._refclim_w1.sqlite                     0.112                           7.24
##  5 B0._refclim_w2.sqlite                     0.126                           7.04
##  6 B0._refclim_w3.sqlite                     0.143                           9.04
##  7 B1_refclim_w1.sqlite                      0.727                          13.4
##  8 B1_refclim_w2.sqlite                      0.688                          13.6
##  9 B1_refclim_w3.sqlite                      0.795                          19.1
## 10 B2_refclim_w1.sqlite                      1.35                           17.7
## 11 B2_refclim_w2.sqlite                      1.40                           19.1
## 12 B2_refclim_w3.sqlite                      1.72                           28.8
## # ... with abbreviated variable name 1: `mean(na.omit(killedVolume/area))`
```

```
# PLOT SECOND Y AXIS IN RELATIVE KILLED VOLUME BY DISTURBANCES IN LANDSCAPE AVG

ylim.bb <- c(0, 600000)
ylim.w <- c(0, 1700000)
```

```
b <- diff(ylim.bb)/diff(ylim.w)
a <- ylim.bb[1] - b*ylim.w[1]

# TO MAKE 2 LINE AND 2 DIFFERENT SCALE PLOT "https://stackoverflow.com/questions/3099219/ggplot-with-2-
ggplot(damage.all,aes(year,barkbeetle/area/tot_vol))+
  geom_col(fill="pink",col="pink")+
  geom_point(aes(y = a + killedVolume/area/tot_vol*b), data = damage.all,size=2) +
  scale_y_continuous(name="Barkbeetle relative damage", sec.axis = sec_axis(~ (. - a)/b,name = "Wind rel
  facet_wrap(~case, ncol=4)+
  theme_bw()
```

```
## Warning: Removed 3480 rows containing missing values (`geom_point()`).
```




```
# SET THE SCALE OF THE Y AXIS. NAME IS THE NAME OF THE LABEL. FIRST Y AXIS SCALE REFERENCE ONE. SEC_AXI
rel_damage <- (damage.all %>%
                 group_by(case) %>%
                 summarise((barkbeetle/tot_vol),
                           (killedVolume/tot_vol)))
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
```

```
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## ‘summarise()‘ has grouped output by ’case’. You can override using the
## ‘.groups‘ argument.
```

```
rel_damage
```

```
## # A tibble: 3,600 x 3
## # Groups:   case [12]
##    case                    ‘(barkbeetle/tot_vol)‘ ‘(killedVolume/tot_vol)‘
##    <chr>                                    <dbl>                    <dbl>
##  1 B0.5_refclim_w1.sqlite                   0.609                       NA
##  2 B0.5_refclim_w1.sqlite                   1.84                        NA
##  3 B0.5_refclim_w1.sqlite                   1.46                        NA
##  4 B0.5_refclim_w1.sqlite                   3.64                        NA
##  5 B0.5_refclim_w1.sqlite                   8.06                        NA
##  6 B0.5_refclim_w1.sqlite                   9.52                        NA
##  7 B0.5_refclim_w1.sqlite                   9.15                        NA
##  8 B0.5_refclim_w1.sqlite                   9.14                        NA
##  9 B0.5_refclim_w1.sqlite                   7.59                        NA
## 10 B0.5_refclim_w1.sqlite                   8.78                        NA
## # ... with 3,590 more rows
```

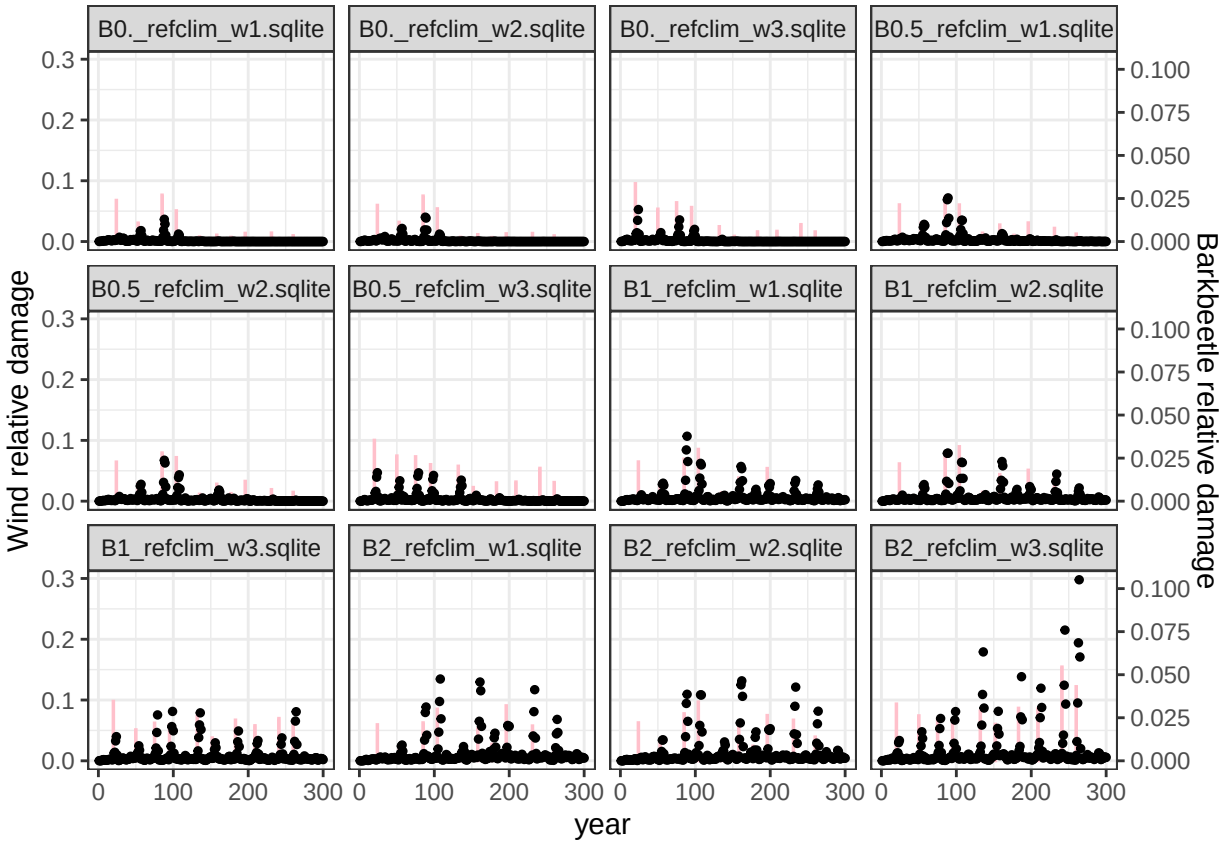
```
relative_damage <- (damage.all %>%
                      group_by(case) %>%
                      summarise(mean(barkbeetle/tot_vol),
                                mean(na.omit(killedVolume/tot_vol))))                    # SUMMARISE TH
relative_damage
```

```
## # A tibble: 12 x 3
##    case                    ‘mean(barkbeetle/tot_vol)‘ mean(na.omit(killedVolume~1
##    <chr>                                        <dbl>                       <dbl>
##  1 B0.5_refclim_w1.sqlite                        18.6                        609.
##  2 B0.5_refclim_w2.sqlite                        18.7                        649.
##  3 B0.5_refclim_w3.sqlite                        20.0                        954.
##  4 B0._refclim_w1.sqlite                          7.81                       516.
##  5 B0._refclim_w2.sqlite                          8.82                       504.
##  6 B0._refclim_w3.sqlite                          9.71                       677.
##  7 B1_refclim_w1.sqlite                          41.1                        763.
##  8 B1_refclim_w2.sqlite                          38.8                        774.
##  9 B1_refclim_w3.sqlite                          45.9                       1140.
## 10 B2_refclim_w1.sqlite                          66.6                        892.
## 11 B2_refclim_w2.sqlite                          70.2                        978.
## 12 B2_refclim_w3.sqlite                          92.3                       1578.
## # ... with abbreviated variable name 1: ‘mean(na.omit(killedVolume/tot_vol))‘
```

```
# variant 1
ylim.bb <- c(0, 600000)
ylim.w <- c(0, 1700000)

b <- diff(ylim.w)/diff(ylim.bb)
```

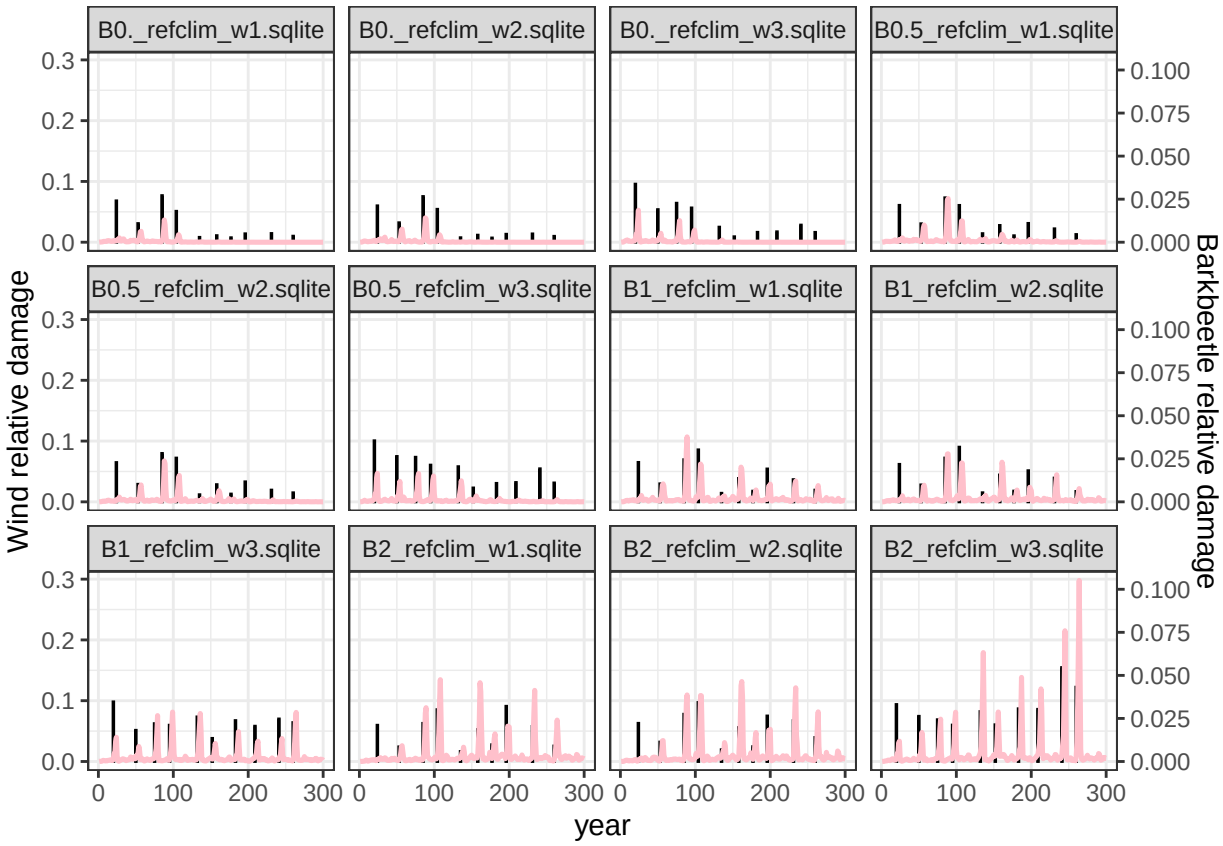
```
a <- ylim.w[1] - b*ylim.bb[1]

# TO MAKE 2 LINE AND 2 DIFFERENT SCALE PLOT "https://stackoverflow.com/questions/3099219/ggplot-with-2-
ggplot(damage.all,aes(year,killedVolume/area/tot_vol))+
  geom_col(fill="pink",col="pink")+
  geom_point(aes(y = a+ barkbeetle/area/tot_vol*b), data = damage.all,size=1) +
  scale_y_continuous(name="Wind relative damage", sec.axis = sec_axis(~ (. - a)/b,name = "Barkbeetle rel
  facet_wrap(~case, ncol=4)+
  theme_bw()
```

```
## Warning: Removed 3480 rows containing missing values (`position_stack()`).
```

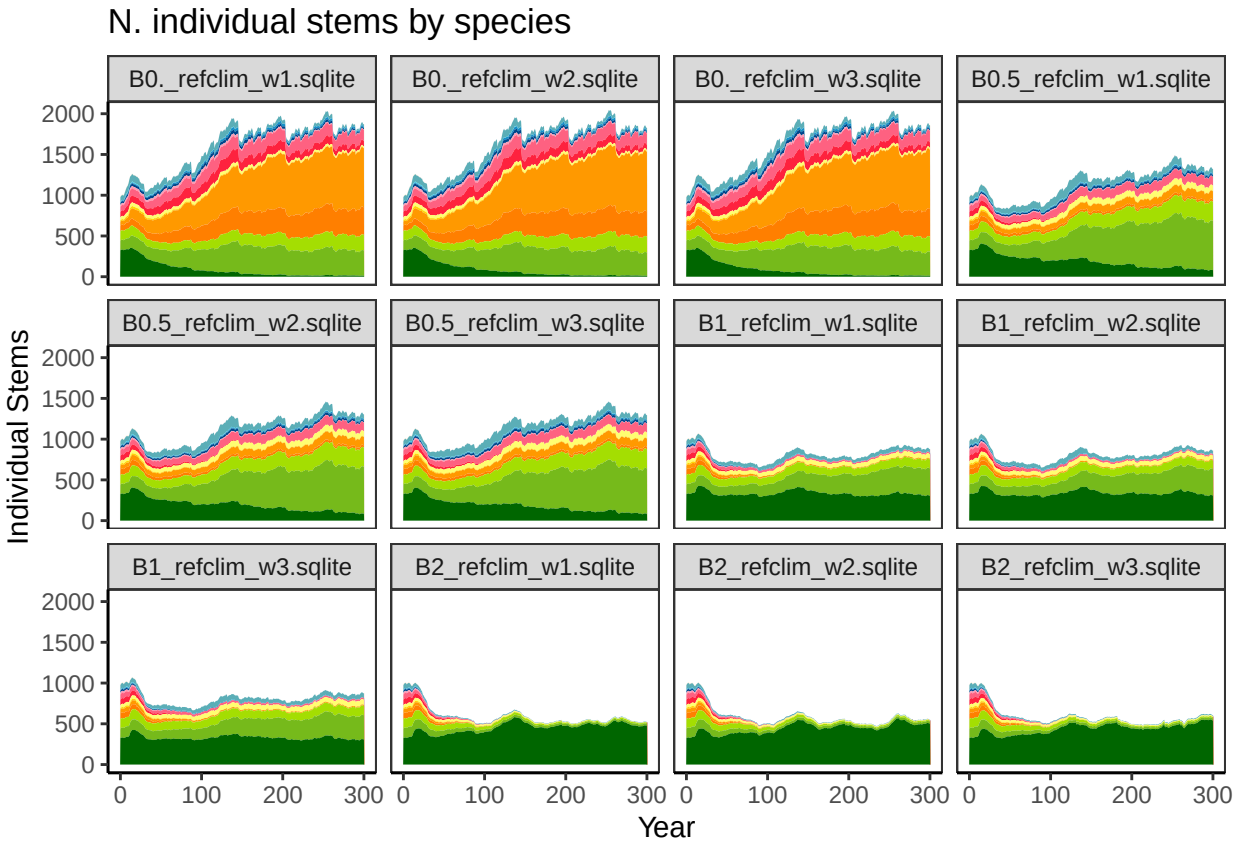



```
# variant 2

b <- diff(ylim.w)/diff(ylim.bb)
a <- ylim.w[1] - b*ylim.bb[1]

# TO MAKE 2 LINE AND 2 DIFFERENT SCALE PLOT "https://stackoverflow.com/questions/3099219/ggplot-with-2-
ggplot(damage.all,aes(year,killedVolume/area/tot_vol))+
  geom_col(fill="grey",col="black")+
  geom_line(aes(y = a+ barkbeetle/area/tot_vol*b), data = damage.all, size=0.9, col="pink") +
  scale_y_continuous(name="Wind relative damage", sec.axis = sec_axis(~ (. - a)/b,name = "Barkbeetle rel
  facet_wrap(~case, ncol=4)+
  theme_bw()
```

```
## Warning: Removed 3480 rows containing missing values (`position_stack()`).
```

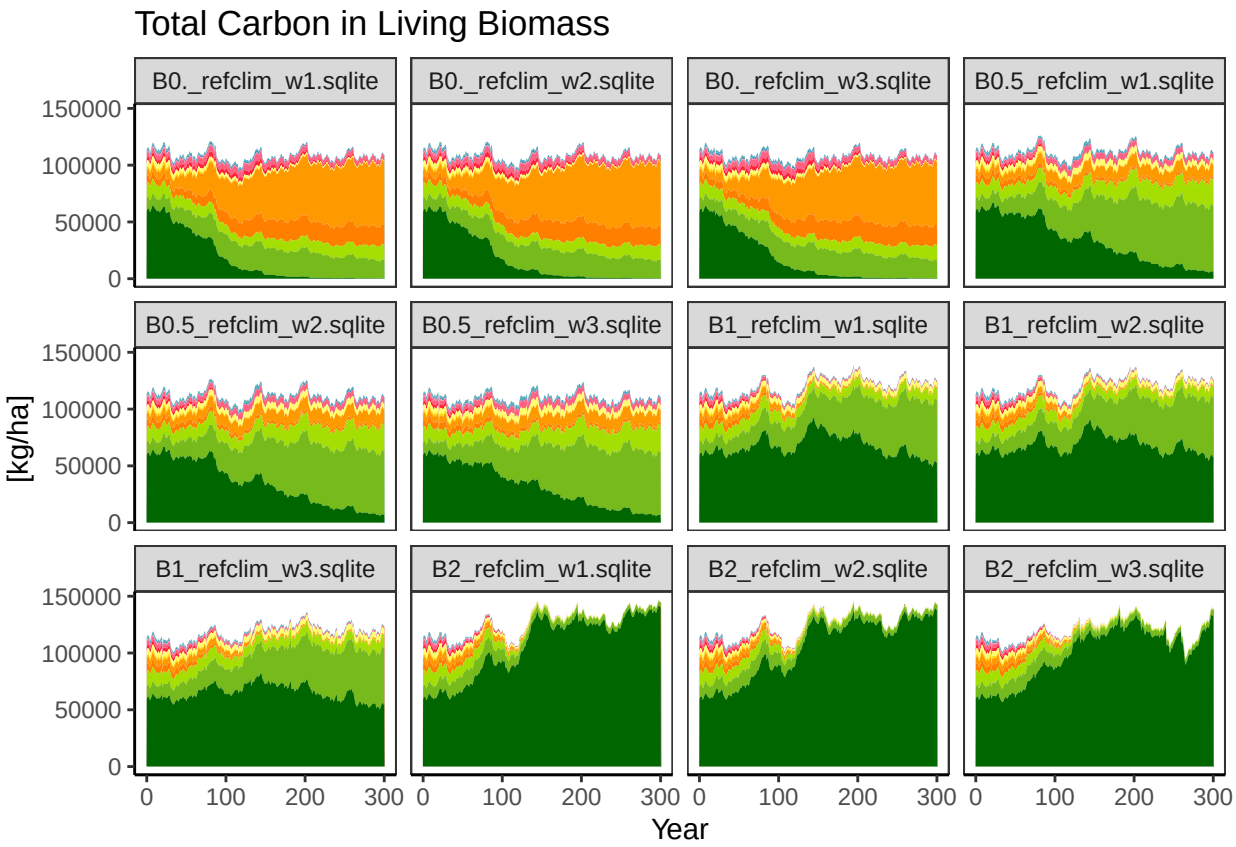



```
#                                   Grid for the paper!                                  #


# PLOT NUMBER OF STEMS GEOM_AREA AT LANDSCAPE LEVEL BY SPECIES


ggplot(lnd, aes(x=year, y=count_ha, fill=factor(species, levels=new_order_gg)))+
  geom_area(show.legend = F)+
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("N. individual stems by species") +
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="Individual Stems", fill = "Species")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()+
  theme(axis.line = element_line(color='black'),
        plot.background = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_blank())
```



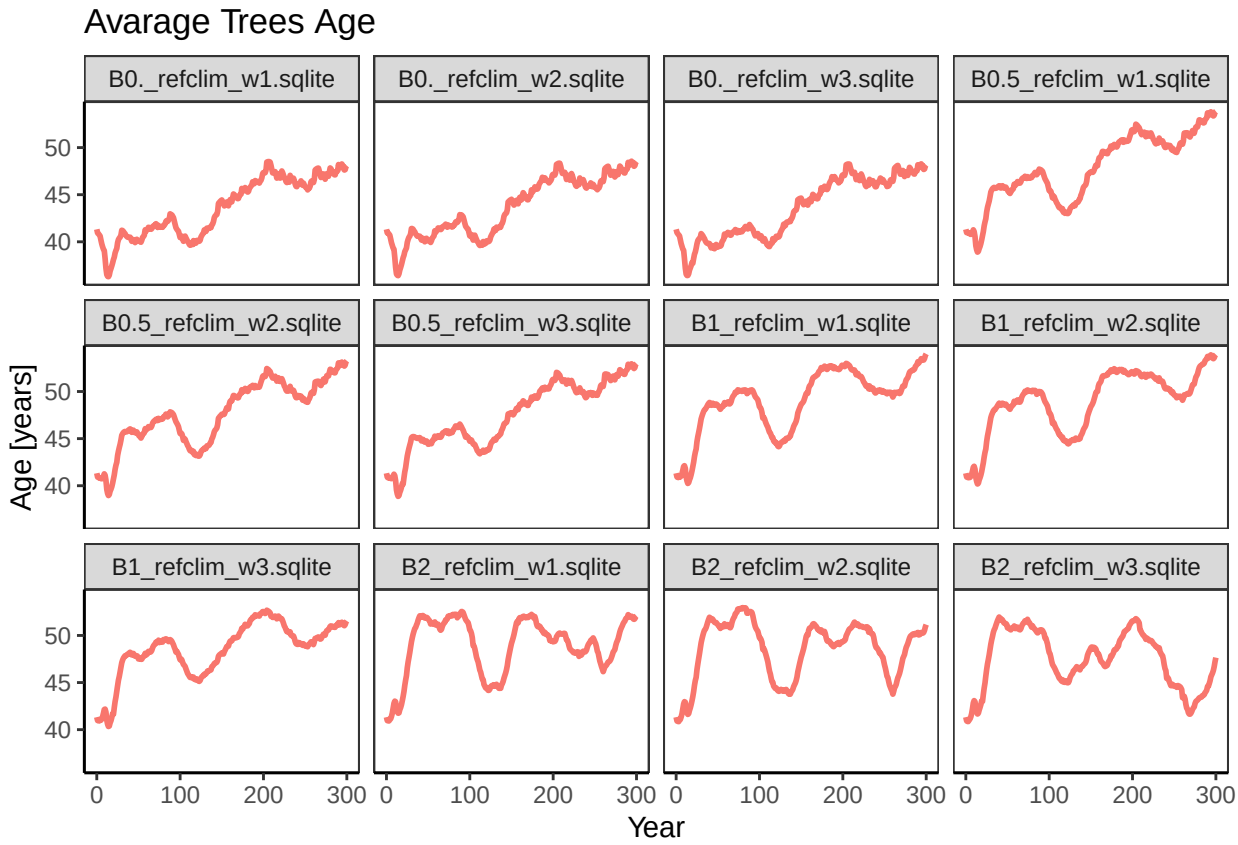

```
#theme_bw()  # this is to not have the gray - classic you will have a blanck on -> in case for others l

# https://www.statology.org/ggplot-remove-gridlines/
```

```
# Total Carbon in Kg (total_carbon_kg   double  total carbon in living biomass (aboveground compartment

ggplot(lnd, aes(year, total_carbon_kg, fill=factor(species, levels=new_order_gg)))+
  geom_area(show.legend = F) +
  scale_fill_manual(values=cols[new_order_gg], guide=guide_legend(reverse=TRUE))+
  ggtitle("Total Carbon in Living Biomass")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="[kg/ha]",fill = "Species")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()+
  theme(axis.line = element_line(color='black'),
        plot.background = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_blank())
```

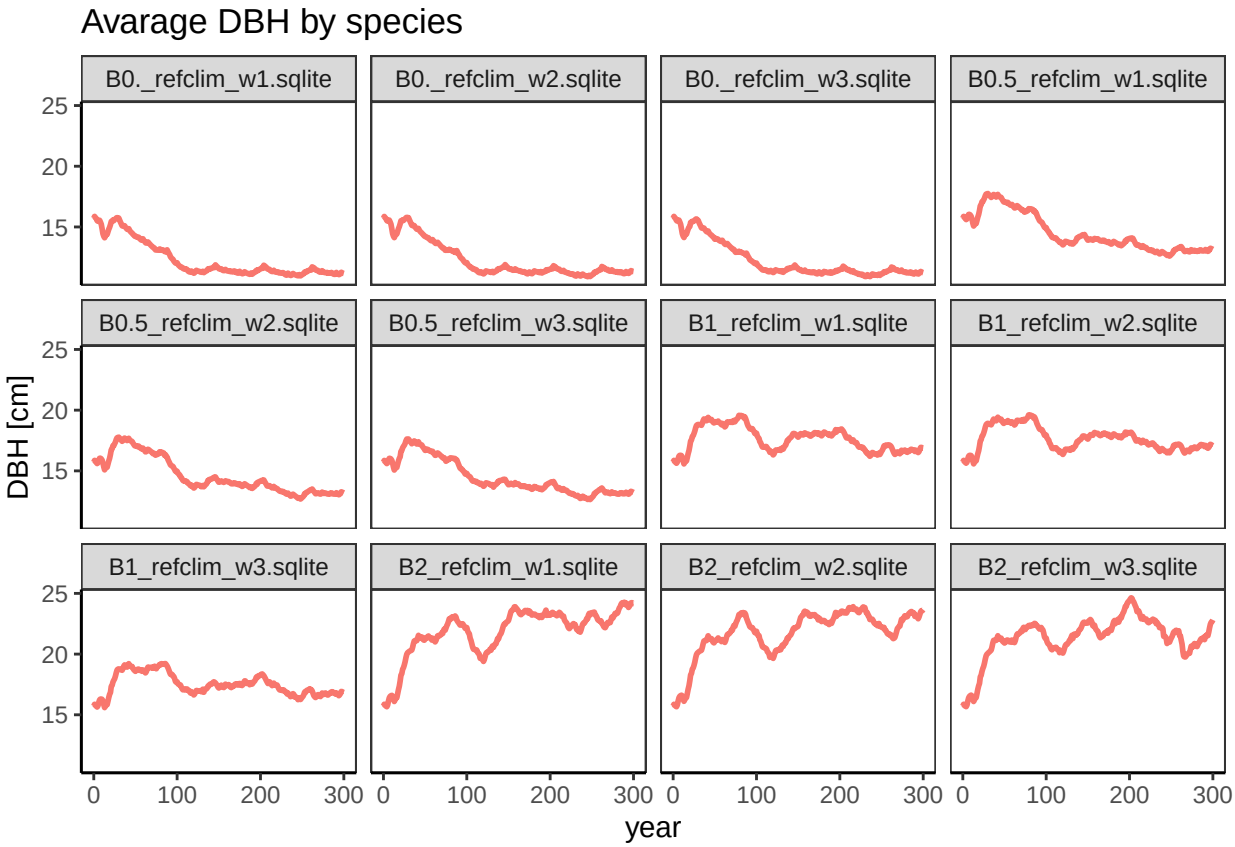



```
# theme_classic()
```

```
# AGE

ggplot(dys, aes(x=year, y=age_mean, color="red"))+
  geom_line(size=1,show.legend = F) +
  ggtitle("Avarage Trees Age")+
  facet_wrap(~run, ncol=4)+
  labs(x = "Year",y="Age [years]")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme_bw()+
  theme(axis.line = element_line(color='black'),
        plot.background = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_blank())
```




```
# PLOT DBH GEOM_AREA AT LANDSCAPE LEVEL AVARAGE ALL SP TOGETHER

ggplot(data=dys, aes(x=year, y=dbh_mean, color=case)) +
  geom_line(size=1,show.legend = F)+
  ggtitle("Avarage DBH by species") +
  facet_wrap(~run, ncol=4)+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("DBH [cm]")+
  theme_bw()+
  theme(axis.line = element_line(color='black'),
        plot.background = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_blank())
```




```
#_____________________________________________________________________________

# NEW KILLED VOLUME CALCULATION INCLUDING BARK BEETLE
# Make the disturbance impact:

# Wind AND bark beetle disturbance impact in the landscape volume in percentages
# wind and bark beetle regime

#_____________________________________________________________________________


killed_volume_w  <- sum(w$killedVolume)
killed_volume_w
```

```
## [1] 29911115
```

```
killed_volume_bb <- sum(bb$killedVolume)
killed_volume_bb
```

```
## [1] 42348794
```

```
killed_volume_dist <- killed_volume_w + killed_volume_bb
killed_volume_dist
```

```
## [1] 72259909
```

```r
killed_volume_per_year_dist  <- killed_volume_dist/300
killed_volume_per_year_dist
```

```
## [1] 240866.4
```

```r
killed_volume_per_year_dist_ha <- killed_volume_per_year_dist/17749.26
killed_volume_per_year_dist_ha
```

```
## [1] 13.5705
```

```r
killed_volume_w  <- sum(wind$killedVolume)
killed_volume_w
```

```
## [1] 5110448
```

```r
killed_volume_bb <- sum(barkbeetle$killedVolume)
killed_volume_bb
```

```
## [1] 9154517
```

```r
killed_volume_dist <- killed_volume_w + killed_volume_bb
killed_volume_dist
```

```
## [1] 14264965
```

```r
killed_volume_per_year_dist  <- killed_volume_dist/200
killed_volume_per_year_dist
```

```
## [1] 71324.83
```

```r
killed_volume_per_year_dist_ha <- killed_volume_per_year_dist/17749.26
killed_volume_per_year_dist_ha
```

```
## [1] 4.018468
```

```r
# FILTER AND GROUP FOR THE NEEDED COLUMNS AND GROUP BY YEAR TO CREATE NEW DATAFRAMES FOR THE % ANALSIS

dfnew1  <- lnd[,c(1,8)]

df_vol = dfnew1 %>% group_by(year)  %>%
  summarise(tot_vol = sum(volume_m3),
            .groups = 'drop')


prop_killed_vol_ha_year <- df_vol %>% mutate(perc.vol=100*killed_volume_per_year_dist_ha/tot_vol)
prop_killed_vol_ha_year
```

```
## # A tibble: 301 x 3
##     year tot_vol perc.vol
##    <int>   <dbl>    <dbl>
##  1     0   3495.    0.115
##  2     1   3589.    0.112
##  3     2   3551.    0.113
##  4     3   3531.    0.114
##  5     4   3408.    0.118
##  6     5   3454.    0.116
##  7     6   3563.    0.113
##  8     7   3652.    0.110
##  9     8   3663.    0.110
## 10     9   3616.    0.111
## # ... with 291 more rows
```

```
summary(prop_killed_vol_ha_year)
```

```
##       year        tot_vol        perc.vol
##  Min.   :  0   Min.   :3015   Min.   :0.1028
##  1st Qu.: 75   1st Qu.:3352   1st Qu.:0.1123
##  Median :150   Median :3469   Median :0.1158
##  Mean   :150   Mean   :3471   Mean   :0.1161
##  3rd Qu.:225   3rd Qu.:3578   3rd Qu.:0.1199
##  Max.   :300   Max.   :3911   Max.   :0.1333
```

```
dfnew1  <- landscape[,c(1,8)]

df_vol = dfnew1 %>% group_by(year)  %>%
  summarise(tot_vol = sum(volume_m3),
            .groups = 'drop')


prop_killed_vol_ha_year <- df_vol %>% mutate(perc.vol=100*killed_volume_per_year_dist_ha/tot_vol)
prop_killed_vol_ha_year
```

```
## # A tibble: 301 x 3
##     year tot_vol perc.vol
##    <int>   <dbl>    <dbl>
##  1     0    291.     1.38
##  2     1    299.     1.34
##  3     2    296.     1.36
##  4     3    294.     1.37
##  5     4    284.     1.41
##  6     5    288.     1.40
##  7     6    297.     1.35
##  8     7    304.     1.32
##  9     8    305.     1.32
## 10     9    301.     1.34
## # ... with 291 more rows
```

```
summary(prop_killed_vol_ha_year)
```

```
##       year         tot_vol         perc.vol
##  Min.   :  0   Min.   :259.1   Min.   :0.9747
##  1st Qu.: 75   1st Qu.:304.8   1st Qu.:1.1002
##  Median :150   Median :336.3   Median :1.1950
##  Mean   :150   Mean   :334.9   Mean   :1.2161
##  3rd Qu.:225   3rd Qu.:365.3   3rd Qu.:1.3182
##  Max.   :300   Max.   :412.3   Max.   :1.5512
```

```
hist(prop_killed_vol_ha_year$perc.vol,
     main = "Landscape proportion of killed volume [m3/ha] by wind and bark beetles per year in CC",
     cex.main = 1, xlab = "[Killed volume / Total volume] = [%]",
     ylab = "Frequency [years]",
     cex.lab = 1,
     col="lightblue",
     breaks = "FD")
```