

2023_FireExperiment_VegStruct.R

Gebruiker

2023-10-22

```
##### study information and libraries #####
```

Fire experiment Loliondo 2023 F Ledidi & H Olf, University of Groningen

contact: h.olf@rug.nl

vegetation composition in 2x2 m plots

developed in R version 4.3.1

Database used: 2023_FireExperiment_Ledidi

You may want to inspect the database used (remove # in next line, will open in browser)

```
# browseURL("https://docs.google.com/spreadsheets/d/1ie3Z00nFTmcYx_UTa0kaRmsZ8FUKSR4CMFdyJ15kiJM/edit?u
```

clear workspace

```
rm(list = ls())
```

restore the libraries to the version used to develop the script

```
# renv::restore()
```

load required libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.3      v readr      2.1.4
```

```
## v forcats    1.0.0      v stringr   1.5.0
```

```
## v ggplot2    3.4.4      v tibble    3.2.1
```

```
## v lubridate  1.9.3      v tidyr     1.3.0
```

```
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(emmeans)
```

```
library(multcomp)
```

```
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
##
## Attaching package: 'TH.data'
##
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
library(multcompView)
```

```
##### read and merge the data #####
```

Show the tables in the database with explanation of contents

```
tables_link<-"https://docs.google.com/spreadsheets/d/e/2PACX-1vSZ62o9ummnSZOY0Z2nwYNT6-xk9fngVpFhdYAIB2~
read_csv(tables_link,show_col_types = F) %>%
  dplyr::select(Table_ID,Contents)
```

```
## # A tibble: 16 x 2
##   Table_ID      Contents
##   <chr>         <chr>
## 1 MetStudyInfo  general information on the study
## 2 MetStill2Do   what still needs to be done on the database
## 3 MetTables     list of table in this database
## 4 MetVariables  explanation of variable names and category codes used in t~
## 5 DimExpPlot    Table of the 1 ha experimental plots, with treatments appl~
## 6 DimVegPlot    Table of the vegetation compositon plots, with coordinates~
## 7 DimClipPlot   Table of the clipping plots
## 8 DimSpecies    Imported table from database SerengetiSpecies, for validat~
## 9 DimExpPlotsCoords Geographic coordinates of the 100 x 100 m experimental plo~
## 10 FactVegPlotObs Vegetation relevee data for the 2x2 m subplots, 3 replicat~
## 11 FactVegStruc  Vegetation structure transect data, recorded every 2 m alo~
## 12 FactTrees     Tree positions and diameter at 30 cm height along a 88 m t~
## 13 FactClippingBiom Clipped biomass data of the repeated clipping of the grazi~
## 14 FactClipPlotVeg Percentage cover per species of the clipping plots prior t~
## 15 FactClipPlotHgt Vegetation height of the clipping plots
## 16 Rscripts      R scripts for the analysis of the data
```

show validated relations between tables in the database, that can be used in, e.g. left_join()

```
relations_link<-"https://docs.google.com/spreadsheets/d/e/2PACX-1vSZ62o9ummnSZOY0Z2nwYNT6-xk9fngVpFhdYAIB2~
read_csv(relations_link, show_col_types = F) |>
  dplyr::filter(relational=="yes") |>
  dplyr::arrange(Variable_ID,Table_ID)
```

```
## # A tibble: 11 x 4
##   Variable_ID Table_ID      relational explanation
##   <chr>        <chr>        <chr>        <chr>
## 1 ClipPlot_ID DimClipPlot    yes          Clipping plot ID, plots are 50 x 5~
## 2 ClipPlot_ID DimExpPlotsCoords yes          Clipping plot ID, plots are 50 x 5~
## 3 ExpPlot_ID   DimClipPlot    yes          Experimental plot (100x100 m) ID
## 4 ExpPlot_ID   DimExpPlot     yes          Experimental plot (100x100 m) ID
## 5 ExpPlot_ID   DimVegPlot     yes          Experimental plot (100x100 m) ID
## 6 SpCode2      DimSpecies     yes          2-letter species code (for large a~
## 7 SpCode6      DimSpecies     yes          6-letter species code
## 8 SpCode6      FactVegPlotObs yes          6-letter species code
## 9 VegPlotObsID FactVegPlotObs yes          vegetation plot observation ID
## 10 VegPlot_ID  DimVegPlot     yes          vegetation plot ID (3 vegetation p~
## 11 VegPlot_ID  FactVegPlotObs yes          vegetation plot ID (3 vegetation p~
```

```
# Read the relevant data tables from the database
```

```
DimExpPlot_link<-"https://docs.google.com/spreadsheets/d/e/2PACX-1vSZ62o9ummnSZOY0Z2nwYNT6-xk9fngVpFhdY
DimExpPlot<- read_csv(DimExpPlot_link, show_col_types = F)
FactVegStruc_link<-"https://docs.google.com/spreadsheets/d/e/2PACX-1vSZ62o9ummnSZOY0Z2nwYNT6-xk9fngVpFhdY
FactVegStruc<- read_csv(FactVegStruc_link, show_col_types = F) |>
  mutate(Date=lubridate::dmy(Date),
         FirstDate=lubridate::dmy(FirstDate))
```

show the unique dates in the dataset

```
unique(sort(FactVegStruc$FirstDate))
```

```
## [1] "2023-02-09" "2023-04-18" "2023-07-25"
```

```
# merge the data tables based on the database relations (common variables)
```

```
VegStruc <- left_join(FactVegStruc,DimExpPlot, by="ExpPlot_ID") |>
  mutate(Treatment=factor(Treatment,levels=c("G","SG","FB","SFB")),
         Treatment=dplyr::recode_factor(Treatment,"G"="grazed,shrubs remained","SG"="grazed,shrubs cleared",
         "FB"="ungrazed, shrubs remained","SFB"="ungrazed,shrubs cleared"),
         SoilCat=dplyr::recode_factor(factor(SoilCat),"black"="black soil","red"="red soil"))
names(VegStruc) # show all variable names in the combined table
```

```
## [1] "Date"          "FirstDate"     "ExpPlot_ID"    "Distance_m"    "VegHgt_cm"
## [6] "DomBelow"      "DomAbove"      "Burned_p"      "PlotCode"      "Block"
## [11] "SoilCat"       "Treatment"     "Grazed"        "Burned"        "BushCleared"
```

```
str(VegStruc)
```

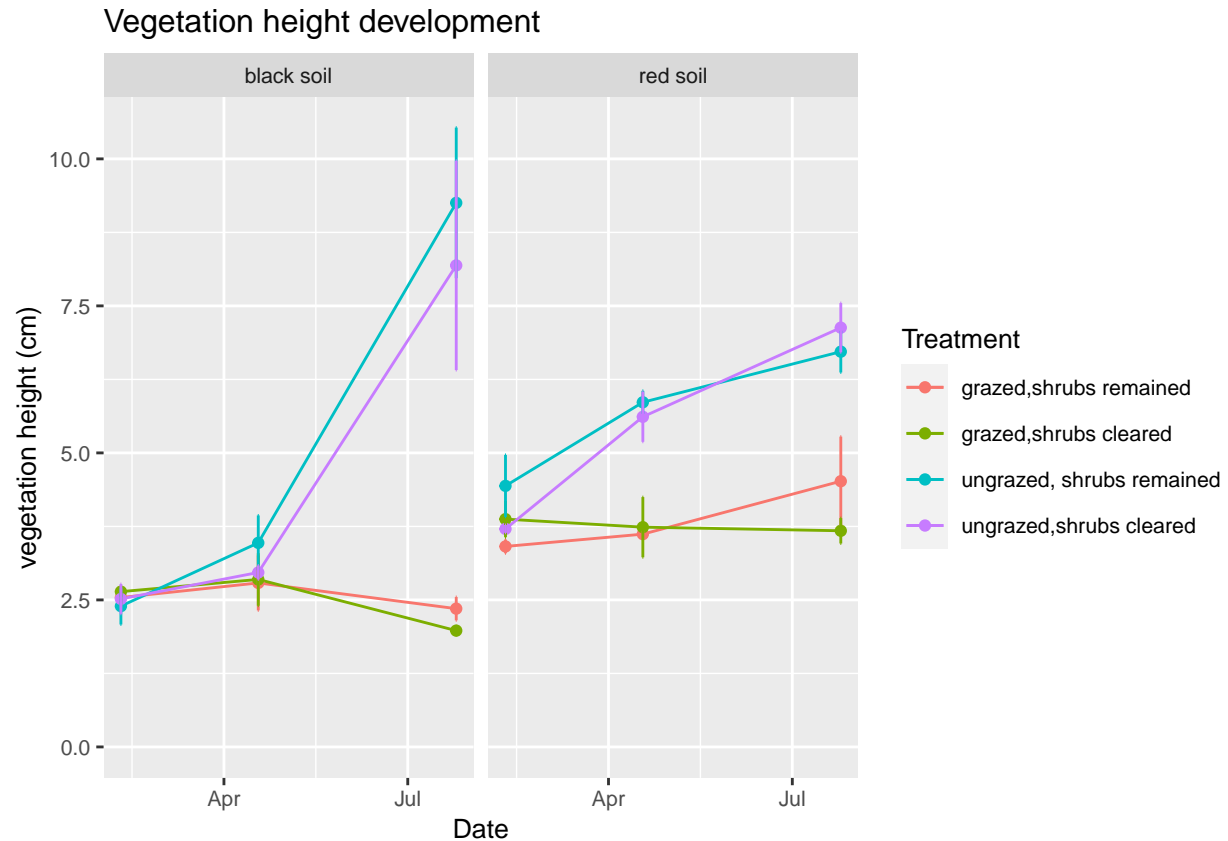
```
## tibble [3,240 x 15] (S3: tbl_df/tbl/data.frame)
## $ Date       : Date[1:3240], format: "2023-02-09" "2023-02-09" ...
## $ FirstDate  : Date[1:3240], format: "2023-02-09" "2023-02-09" ...
## $ ExpPlot_ID : chr [1:3240] "KA2" "KA2" "KA2" "KA2" ...
## $ Distance_m : num [1:3240] 6 8 10 12 14 16 18 20 22 24 ...
## $ VegHgt_cm  : num [1:3240] 2 2 1.5 2.5 2.5 2 1.5 1.5 1 2.5 ...
## $ DomBelow   : chr [1:3240] "grass" "grass" "grass" "grass" ...
## $ DomAbove   : chr [1:3240] "none" "none" "none" "woody" ...
```

```
## $ Burned_p      : num [1:3240] 0 0 0 0 0 0 0 0 0 0 ...
## $ PlotCode      : chr [1:3240] "KA2-FB" "KA2-FB" "KA2-FB" "KA2-FB" ...
## $ Block         : chr [1:3240] "A" "A" "A" "A" ...
## $ SoilCat       : Factor w/ 2 levels "black soil","red soil": 1 1 1 1 1 1 1 1 1 1 ...
## $ Treatment     : Factor w/ 4 levels "grazed,shrubs remained",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ Grazed        : chr [1:3240] "no" "no" "no" "no" ...
## $ Burned        : chr [1:3240] "yes" "yes" "yes" "yes" ...
## $ BushCleared   : chr [1:3240] "no" "no" "no" "no" ...
```

```
##### summarize and plot the data #####
# Summarize and plot the mean vegetation height per treatment over time
fig01<-VegStruct %>%
  group_by(ExpPlot_ID, SoilCat,Treatment,FirstDate) |>
  # calculate first the mean height per plot per block
  summarize(VegHgt_cm=mean(VegHgt_cm,na.rm=T)) |>
  group_by(SoilCat,Treatment,FirstDate) |>
  # calculate the mean height per treatment and soil category
  summarize(mean=mean(VegHgt_cm,na.rm=T),
             sd=sd(VegHgt_cm,na.rm=T),
             n=n(),
             se=sd/sqrt(n)) |>
  arrange(FirstDate,SoilCat,Treatment) |>
  ggplot(data=_, aes(x=FirstDate,y=mean,col=Treatment)) +
  # scale_fill_manual(values=c("#757575","#ef756a")) +
  geom_line(linewidth=0.5) +
  geom_point(size=1.5) +
  geom_errorbar(aes(ymin = mean-se, ymax = mean+se),
                width=0.5,linewidth=0.5) +
  xlab('Date') +
  ylab('vegetation height (cm)') +
  ylim(0,NA) +
  ggtitle('Vegetation height development') +
  facet_wrap(~SoilCat) +
  theme(text = element_text(size=10))
```

```
## 'summarise()' has grouped output by 'ExpPlot_ID', 'SoilCat', 'Treatment'. You
## can override using the '.groups' argument.
## 'summarise()' has grouped output by 'SoilCat', 'Treatment'. You can override
## using the '.groups' argument.
```

```
fig01 # show resulting figure
```



save the figure to a png file at good resolution

```
ggsave(filename="./figures/fig01.png",plot=fig01,width=1920, height=1080, units='px')
```

```
# show dates
```

```
unique(VegStruct$FirstDate)
```

```
## [1] "2023-02-09" "2023-04-18" "2023-07-25"
```

```
##### statistical tests #####
```

test the significance of differences between the treatments on the last date using a three-way ANOVA

```
model01<-VegStruct |>
  filter(FirstDate=="2023-07-25") |>
  lm(VegHgt_cm~SoilCat*Grazed*BushCleared,data=_)
anova(model01)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: VegHgt_cm
```

```
##
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## SoilCat	1	0.9	0.9	0.0983	0.75400
## Grazed	1	5989.2	5989.2	684.8795	< 2.2e-16 ***
## BushCleared	1	9.5	9.5	1.0904	0.29663

```
## SoilCat:Grazed          1  949.2   949.2 108.5415 < 2.2e-16 ***
## SoilCat:BushCleared     1   28.9    28.9   3.3052   0.06934 .
## Grazed:BushCleared      1  184.4   184.4  21.0848  4.914e-06 ***
## SoilCat:Grazed:BushCleared 1    2.8     2.8   0.3229   0.57001
## Residuals              1073 9383.3    8.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#### test the differences with a Tukey test
```

calculate a variable representing the groups to be compared

```
VegStruct <- VegStruct |>
  mutate(groups=interaction(SoilCat,Treatment))
```

compute one-way anova on this variable

```
model02<-VegStruct |>
  filter(FirstDate=="2023-07-25") |>
  lm(VegHgt_cm~groups,data=_)
anova(model02)
```

```
## Analysis of Variance Table
##
## Response: VegHgt_cm
##          Df Sum Sq Mean Sq F value    Pr(>F)
## groups      7 7078.1 1011.15  114.57 < 2.2e-16 ***
## Residuals 1073 9470.1    8.83
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

calculate tukey tests

```
tukey <- emmeans::emmeans(model02, "groups", type = "response")
multcomp::cld(tukey, Letter="abcdefg")
```

```
## groups          emmean    SE    df lower.CL upper.CL
## black soil.grazed,shrubs cleared      1.98 0.256 1073      1.48      2.48
## black soil.grazed,shrubs remained      2.35 0.255 1073      1.85      2.85
## red soil.grazed,shrubs cleared         3.67 0.255 1073      3.17      4.17
## red soil.grazed,shrubs remained         4.52 0.256 1073      4.02      5.02
## red soil.ungrazed, shrubs remained       6.72 0.257 1073      6.22      7.22
## red soil.ungrazed,shrubs cleared        7.13 0.256 1073      6.63      7.63
## black soil.ungrazed,shrubs cleared       8.24 0.256 1073      7.74      8.74
## black soil.ungrazed, shrubs remained     9.25 0.256 1073      8.75      9.75
## .group
## a
## a
## b
## b
## c
## c
```

```

##      d
##      d
##
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 8 estimates
## significance level used: alpha = 0.05
## NOTE: If two or more means share the same grouping symbol,
##       then we cannot show them to be different.
##       But we also did not show them to be the same.

```

this means that four groups are different, while bush clearing within grazing and bush clearing within ungrazed is not significantly different