dataOverview

Pieter Clauw

03/06/2019

```r
library(googlesheets)
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.

##

## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.

##
## Attaching package: 'gdata'

## The following object is masked from 'package:stats':
##
##     nobs

## The following object is masked from 'package:utils':
##
##     object.size

## The following object is masked from 'package:base':
##
##     startsWith
```

```r
library(nlme)
library(emmeans)
library(splines)
source('functions.r')
```

```r
knitr::opts_knit$set(root.dir = "/Volumes/nordborg/user/pieter.clauw/Documents/Experiments/FieldRNA/")
lemna.aug <- read.csv('/Volumes/nordborg/user/pieter.clauw/Documents/Experiments/FieldRNA/Data/Growth/In
lemna.sep <- read.csv('/Volumes/nordborg/user/pieter.clauw/Documents/Experiments/FieldRNA/Data/Growth/In
lemna.oct <- read.csv('/Volumes/nordborg/user/pieter.clauw/Documents/Experiments/FieldRNA/Data/Growth/In
# get meta information on each position
potInfo.gs <- gs_key("11ttBZsMiamn4zEDHUNT-L8CsK4EoAwtbSSUjTs5YEj8")
```

```
## Sheet successfully identified: "Field _Setup"
```

```r
potInfo <- as.data.frame(gs_read(potInfo.gs, ws = 'Randomisation'))
```

```
## Accessing worksheet titled 'Randomisation'.

## Parsed with column specification:
## cols(
##   ID = col_character(),
##   block = col_character(),
##   tray = col_double(),
##   trayRow = col_character(),
##   trayColumn = col_double(),
##   acn = col_character(),
##   month = col_character(),
##   day = col_character(),
##   time = col_character(),
##   rep = col_double(),
##   acnCode = col_double(),
##   comment = col_character(),
##   died = col_character(),
##   harvest = col_character(),
##   flower = col_double(),
##   Budding = col_double(),
##   Seed = col_character(),
##   backUp = col_character()
## )
```

```
# get px/mm2 scael for each picture
distance.xls <- read.xls('/Volumes/nordborg/user/pieter.clauw/Documents/Experiments/FieldRNA/Data/Growt
emptyPots <- read.table('/Volumes/nordborg/user/pieter.clauw/Documents/Experiments/FieldRNA/Data/Growth
# get info on autumn temperature clusters
autumnTempClust <- read.table('/Volumes/nordborg/user/pieter.clauw/Documents/Experiments/QandD2/Results

# relative growth rates of simulated Swedish autumn
files <- list.files(path = '/Volumes/nordborg/user/pieter.clauw/Documents/Experiments/QandD2/Data/Growt
RGR.GC <- lapply(files, read.csv)
names(RGR.GC) <- unlist(lapply(files, function(path){strsplit(path, "\\/|\\_|\\.")[[1]][15]}))
```

emptyPots are pots that did not contain a plant on the respective date that the picture was taken as seen on the respective pictures. The empty pots resemble both pots left empty on purpose, plants that died, plants that were harvested.

```
design <-data.frame('TrayNumber' = seq(1,36), 'Block' = c(rep('A', 12), rep('B', 12), rep('C', 12)), 'T
design$Tray <- paste(design$Block, design$TrayBlockNr, sep = '')
trays <- design$Tray

dates <- data.frame('date' = c(26082018, 17092018, 16102018, 13112018, 19042019), 'month' = factor(c('a
time0 <- strptime(26082018, format = "%d%m%Y")
dates$days <- unlist(lapply(dates$date, function(date)
  {
    date <- strptime(date, format = "%d%m%Y")
    timepoint <- round(as.numeric(difftime(date, time0, units = 'days')))
    return(timepoint)
    }))

accessions <- unique(potInfo$acn)
accessions <- accessions[accessions != 'empty_pot']

# accession visualisation
acnVis <- data.frame('name' = accessions, 'acnColour' = rainbow(n = length(accessions), s = 1, v = 1))
acnVis$accession <- unlist(lapply(acnVis$name, function(name){strsplit(name, '_')[[1]][1]}))
# include colouring for local autumn temperature clusters
autumnTempClustColours <- rainbow(6, s = 0.8, v = 1, start = 0.05, end = 0.78)
autumnTempClust$tmeanAutumnClustTemp <- factor(autumnTempClust$tmeanAutumnClustTemp, levels = c('4C', '
autumnTempClust$autumnTempCol <- autumnTempClustColours[autumnTempClust$tmeanAutumnClustTemp]
acnVis$tmeanAutumnTempClustTemp <- autumnTempClust$tmeanAutumnClustTemp[match(acnVis$accession, autumnT
acnVis$autumnTempCol <- autumnTempClust$autumnTempCol[match(acnVis$accession, autumnTempClust$acn)]
# marker used as reference is 14.1cm
marker <- 14.1
```

TODO

change accesison into name where appropriate accession should change into accession ID everywhere

```
# tray A4 in september has one correct and one incorrect image analysis
# remove the image analysis with timestamp 14.05.2019 at 13:28:44
lemna.sep <- lemna.sep[lemna.sep$Analysis.Time.Stamp != "14.05.2019 13:28:44", ]
# combine data
#lemna <- rbind(lemna.aug, lemna.sep, lemna.oct)
lemna <- rbind(lemna.aug, lemna.sep, lemna.oct)
lemna <- lemna[ ,c('ROI.Label', 'Snapshot.ID.Tag', 'ROI.Object.Sum.Area', 'Caliper.Length', 'Circumferen
lemna$pot <- unlist(lapply(lemna$ROI.Label, function(ROI)
  {
    row <- tolower(substr(ROI, 1, 1))
    col <- as.numeric(substr(ROI, 2, 3))
    pot <- paste(row, col, sep = '')
    return(pot)
  }))
# Snapshot.ID.Tag contains date and tray number
lemna$Tray <- unlist(lapply(lemna$Snapshot.ID.Tag, function(ID)
  {
    trayNr <- strsplit(ID, split = '_')[[1]][1]
    tray <- design$Tray[design$TrayNumber == trayNr]
    return(tray)
  }))
lemna$block <- unlist(lapply(lemna$Tray, function(tray){substr(tray,1,1)}))
lemna$position <- paste(lemna$Tray, lemna$pot, sep = '_')
lemna$ID <- paste(lemna$Snapshot.ID.Tag, lemna$pot, sep = '_')
lemna$date <- unlist(lapply(lemna$Snapshot.ID.Tag, function(ID){strsplit(ID, split = '_')[[1]][2]}))
lemna$month <- dates$month[match(lemna$date, dates$date)]

# multiple ROI.objects. Take only first measurement of non-zero measurements and only ROI.Object.Sum.Ar
objectCount <- table(lemna$ID)
IDmulti <- names(objectCount[objectCount > 1])

for (ID in IDmulti)
{
  ID.objectCnt <- objectCount[ID]
  # check  which ones are zero and remove, decide on plants left to make NA
  #zeroNr <- nrow(lemna[lemna$ID == ID & lemna$ROI.Object.Sum.Area == 0, ])
  #if (zeroNr == ID.objectCnt){print(paste('only zero measurement for', ID, 'will be removed through em
  # amount of plants that need to be removed apsart from zero area plants
  #NAIDnr <- ID.objectCnt - zeroNr
  # remove additional ROI.objects
  #if (NAIDnr > 1){lemna$ROI.Object.Sum.Area[lemna$ID == ID][2:ID.objectCnt] <- NA}
  lemna$ROI.Object.Sum.Area[lemna$ID == ID][2:ID.objectCnt] <- NA
  # remove extra measurement with zero area
  #if (zeroNr > 0){lemna$ROI.Object.Sum.Area[lemna$ID == ID & lemna$ROI.Object.Sum.Area == 0] <- NA}
}
# remove multiple ROI.objects
lemna <- lemna[!is.na(lemna$ROI.Object.Sum.Area), ]




# ROI.Label is the plant identifier
```

```r
# transform pixels to cm2
# get conversion ratios per tray
distance.xls$cm[distance.xls$cm == 'marker'] <- marker
distance.xls$cm <- as.numeric(distance.xls$cm)
conversionRates <- data.frame('Date' = integer(), 'Tray' = character(), 'Columns' = character(), 'Conve
for (i in 1:nrow(distance.xls))
{
  if (nchar(distance.xls$Trays[i]) < 1){next()}
  tray <- trimws(strsplit(distance.xls$Trays[i], split = ';')[[1]])
  columns <- distance.xls$columns[i]
  date <- distance.xls$date[i]
  pxls <- distance.xls$pixels[i]
  cms <- distance.xls$cm[i]
  conversion <- (cms/pxls)^2
  # for august the converison rates of the composite images (merge of two pictures of half a tray)
  if (date == '26082018' & grepl('DSC', distance.xls$original.file[i])){next()}

  convDF <- data.frame('Date' = rep(date, length(tray)), 'Tray' = tray, 'Columns' = rep(columns, length
  conversionRates <- rbind(conversionRates, convDF)
}

# convert pixels to mm2
lemna$AreaCm <- apply(lemna, MARGIN = 1, FUN = convertPxlToCm2, conversionRates = conversionRates)
lemna$AreaMm <- lemna$AreaCm * 10
lemna$logAreaMm <- log(lemna$AreaMm)
```

Plants were harvested every month. Remove positions that were empty in the taken pictures. (starting from October)

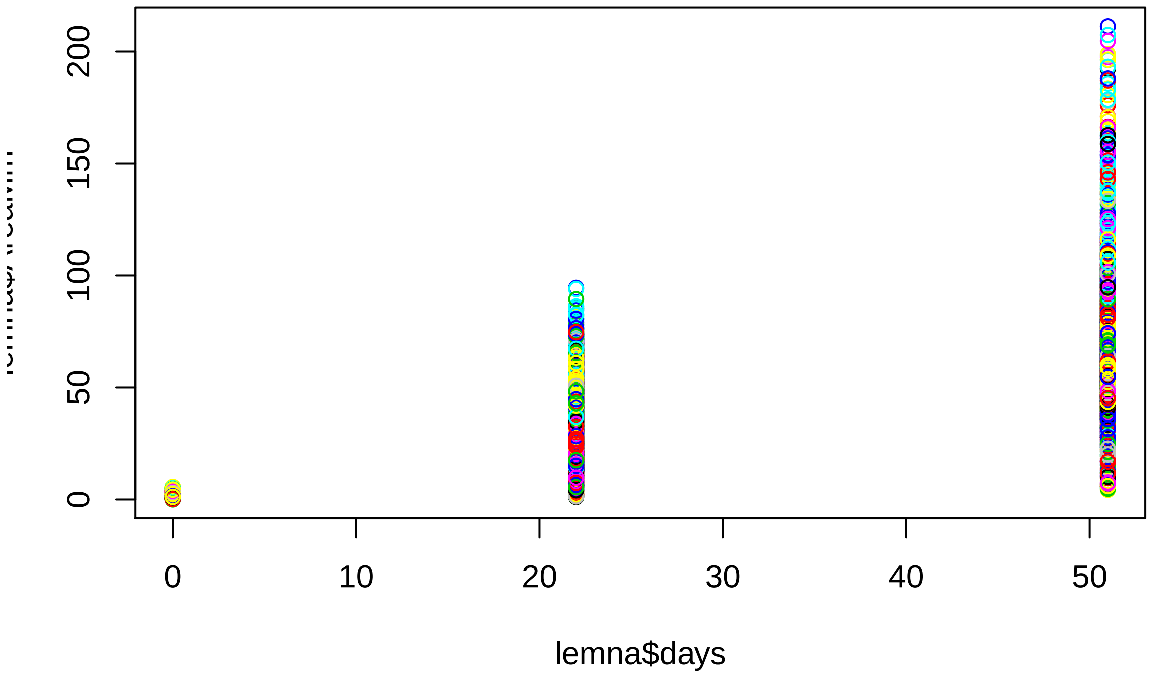
```r
emptyPots$Pot <- unlist(lapply(emptyPots$Pot, function(pot)
{
  row <- tolower(substr(pot, 1,1))
  col <- substr(pot, 2, nchar(pot))
  #col <- formatC(as.numeric(col), width = 2, flag = '0')
  return(paste(row, col, sep = ''))
}))

for (i in 1:nrow(emptyPots))
{
  date <- emptyPots$Date[i]
  tray <- emptyPots$Tray[i]
  pot <- emptyPots$Pot[i]
  lemna$AreaCm[lemna$Tray == tray & lemna$date == date & lemna$pot == pot] <- NA
}
lemna <- lemna[!is.na(lemna$AreaCm), ]

# check for pots with zero values
lemna$ID[lemna$AreaCm == 0]
```

```
## character(0)
```

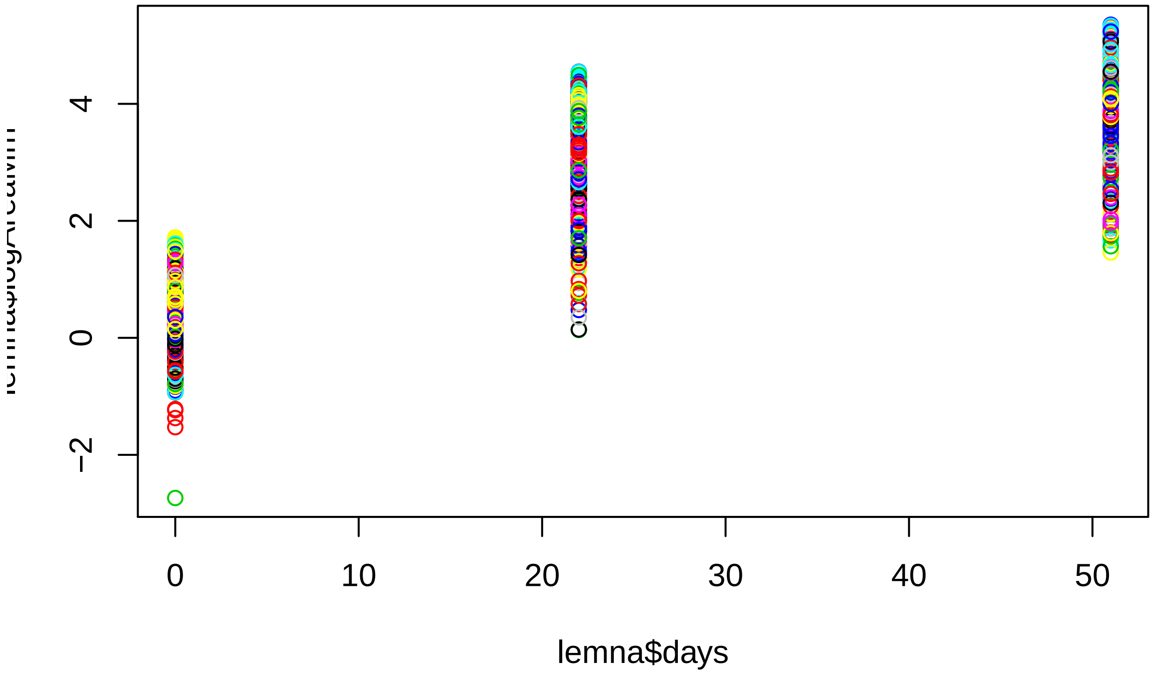
```r
lemna$accession <- potInfo$acn[match(lemna$position, potInfo$ID)]
```

```r
# get timepoints as days
lemna$days <- dates$days[match(lemna$date, dates$date)]
```

```
plot(lemna$days, lemna$AreaMm, col = as.factor(lemna$accession))
```



```
plot(lemna$days, lemna$logAreaMm, col = as.factor(lemna$accession))
```



From teh plot it looks like for the logAreaMm, a polynomial model will be a better fit than a linear or quadratic model. Model with time as factor is much more heavy and does not give a better fit per se.

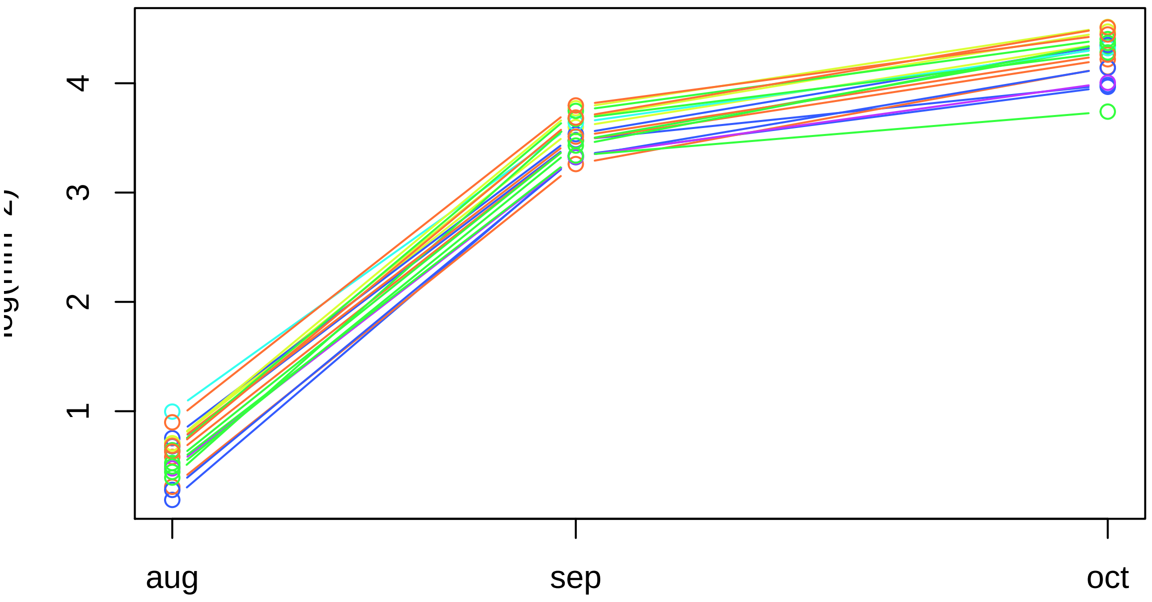
```
# model with time as numeric
lmm.time <- lme(logAreaMm ~ days*accession, random = ~ 1 + days|block/position, data = lemna)
#lmm.polyTime <- lme(logAreaMm ~ (days + I(days^2) + I(days^3))*accession, random = ~ 1 + days|block/po
#lmm.polyTime <- lme(logAreaMm ~ (days + I(days^2))*accession, random = ~ 1 + days|block/position, data
# settng lmeControl to 'old' optimastion algorithm prevents lme to run out of iterations.
ctrl <- lmeControl(opt='optim')
lmm.spline <- lme(logAreaMm ~ bs(days, degree= 1, knots = 22) * accession, random = ~ 1 + days|block/po

# correct for heteroscedascity???
```

polynomial model requires one more timepoint than number of tested parameters. polynomial to the power 3 requires 4 timepoints. (Wait for november data). quadratic model does not converge.

linear spline model has lower AIC (`{r AIC(lmm.spline)}`) compared to linear model withou spline (`{r AIC(lmm.time)}`)

```
# create newdata for emm
nrTimepoints <- 3
predictionData <- data.frame('accession' = rep(accessions, nrTimepoints), 'days' = rep(dates$days[1:nrT

adjM <- emmeans(lmm.spline, ~ days * accession, cov.reduce = F, data = predictionData)
s <- summary(adjM)
emm <- data.frame('days' = s$days, 'accession' = s$accession, 'logAreaMm' = s$emmean, 'SE' = s$SE, 'df'
```

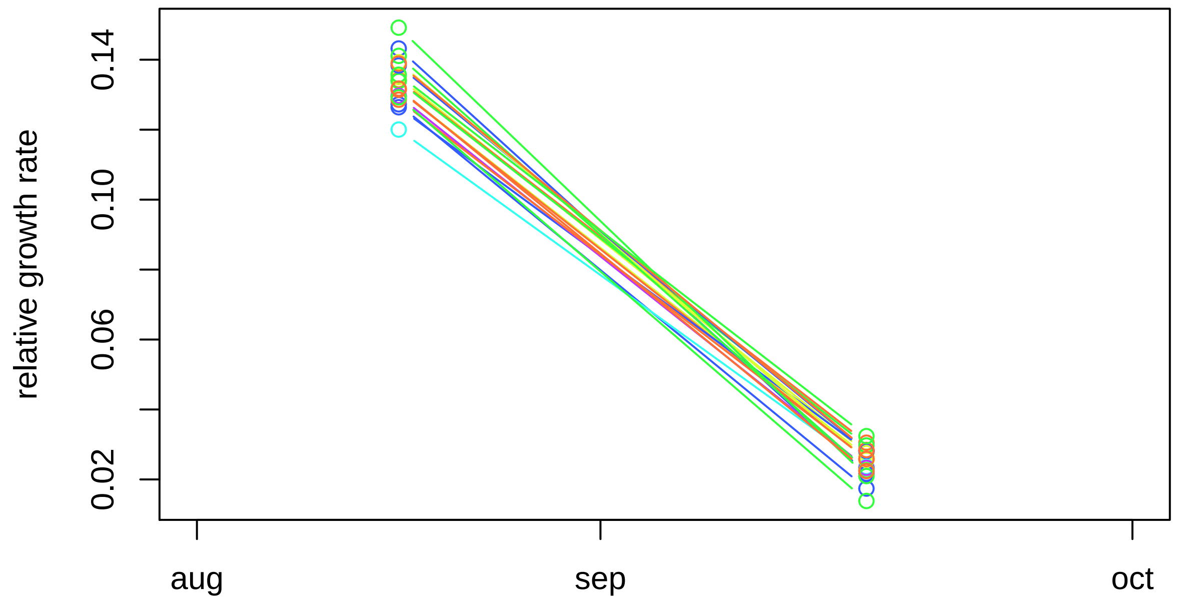
```
minLogArea <- min(emm$logAreaMm)
maxLogArea <- max(emm$logAreaMm)
minDays <- min(emm$days)
maxDays <- max(emm$days)
plot(NA,NA, xlim = c(minDays, maxDays), ylim = c(minLogArea, maxLogArea), xaxt = 'n', xlab = '', ylab =
for (acn in accessions)
{
  emm.a <- emm[emm$accession == acn, ]
  lines(emm.a$days, emm.a$logAreaMm, type = 'b', col = acnVis$autumnTempCol[acnVis$accession == strspli
}
# xaxis with months
axis(side = 1, at = unique(emm$days), labels = dates$month[1:nrTimepoints])
```



```
#legend('topleft', legend = acnVis$accession)
```

```r
RGR <- data.frame('accession' = character(), 'days' = numeric(), 'rgr' = numeric())

days <- unique(emm$days)
for (i in 2:length(days))
{
  for (acn in accessions)
  {
    day1 <- days[i-1]
    day2 <- days[i]
    emm.time.acn <- emm[emm$days %in% c(day1, day2) & emm$accession == acn, ]
    mod <- lm(logAreaMm ~ days, data = emm.time.acn)
    rgr <- mod$coefficients[2]
    period <- paste(dates$month[dates$days == day1], dates$month[dates$days == day2], sep = '_')

    RGR.time.acn <- data.frame('accession' = acn, 'days' = mean(c(day1, day2)), 'period' = period, 'rgr
    RGR <- rbind(RGR, RGR.time.acn)
  }
}
```

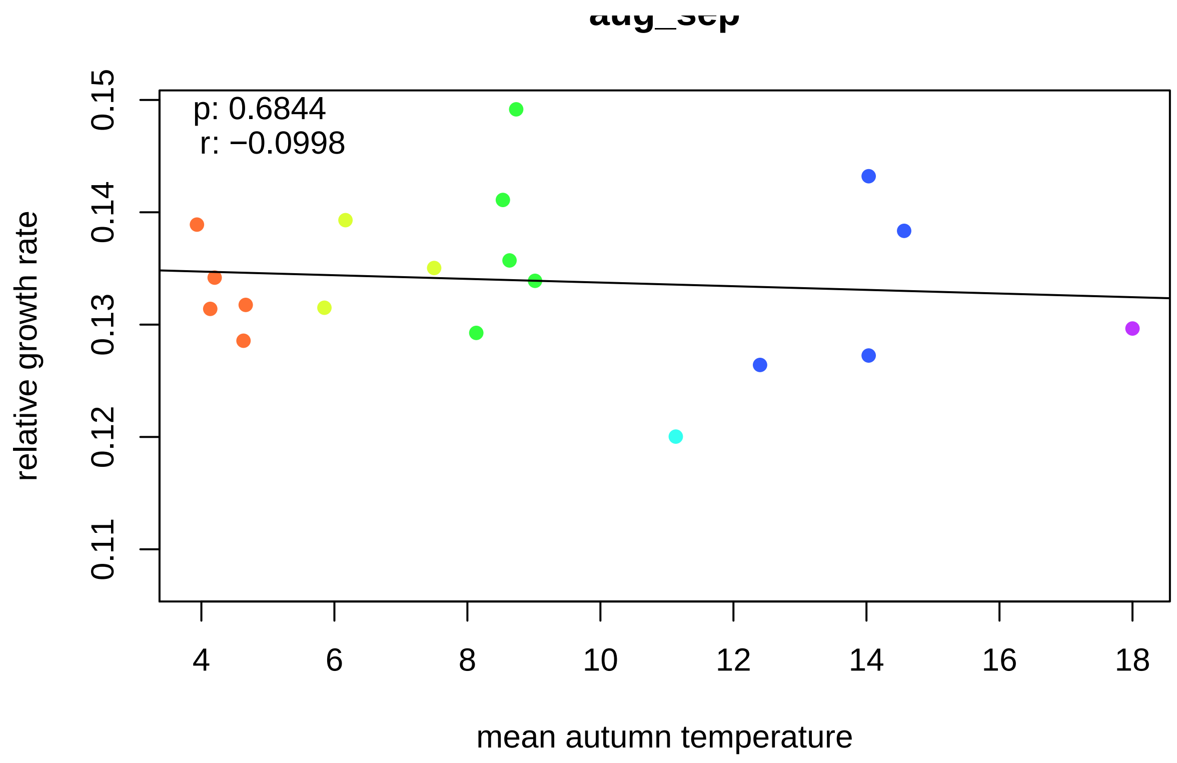
```r
minRGR <- min(RGR$rgr)
maxRGR <- max(RGR$rgr)
plot(NA,NA, xlim = c(minDays, maxDays), ylim = c(minRGR, maxRGR), xaxt = 'n', xlab = '', ylab = "relati
for (acn in accessions)
{
  RGR.a <- RGR[RGR$accession == acn, ]
  lines(RGR.a$days, RGR.a$rgr, type = 'b', col = acnVis$autumnTempCol[acnVis$accession == strsplit(acn,
}
# xaxis with months
axis(side = 1, at = unique(emm$days), labels = dates$month[1:nrTimepoints])
```

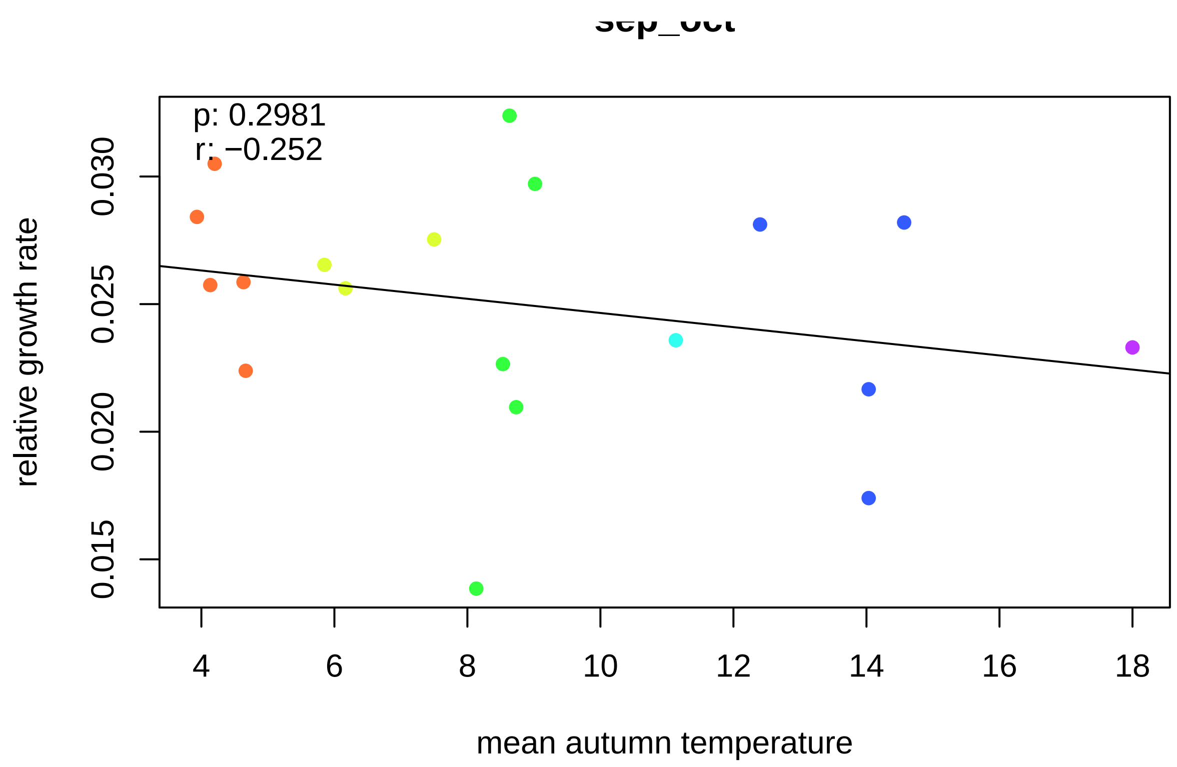


```r
#legend('topleft', legend = acnVis$accession)
```

Correlate relative grotw rates with mean local autumn temperature

```
RGR$acn <- unlist(lapply(RGR$accession, function(x){strsplit(x, '_')[[1]][1]}))
RGR$autumnTemp <- autumnTempClust$meanAutumn[match(RGR$acn, autumnTempClust$acn)]/10
RGR$autumnTempCol <- acnVis$autumnTempCol[match(RGR$acn, acnVis$accession)]
minAutumnTemp <- min(RGR$autumnTemp, na.rm = T)
maxAutumnTemp <- max(RGR$autumnTemp, na.rm = T)

#plot(NA,NA, xlim = c(minAutumnTemp, maxAutumnTemp), ylim = c(minRGR, maxRGR))
for(period in unique(RGR$period))
{
  RGR.p <- RGR[RGR$period == period, ]
  plot(RGR.p$autumnTemp, RGR.p$rgr, col = RGR.p$autumnTempCol, pch = 16, xlab = 'mean autumn temperature
  abline(lm(RGR.p$rgr ~ RGR.p$autumnTemp))
  ct <- cor.test(RGR.p$autumnTemp, RGR.p$rgr)
  text(x = par('usr')[1], y = par('usr')[4], paste('p: ', format.pval(ct$p.value, digits = 3, eps = 0.00
  text(x = par('usr')[1], y = par('usr')[4], paste('r: ', round(ct$estimate, 4), sep = '' ), adj = c(-0
}
```

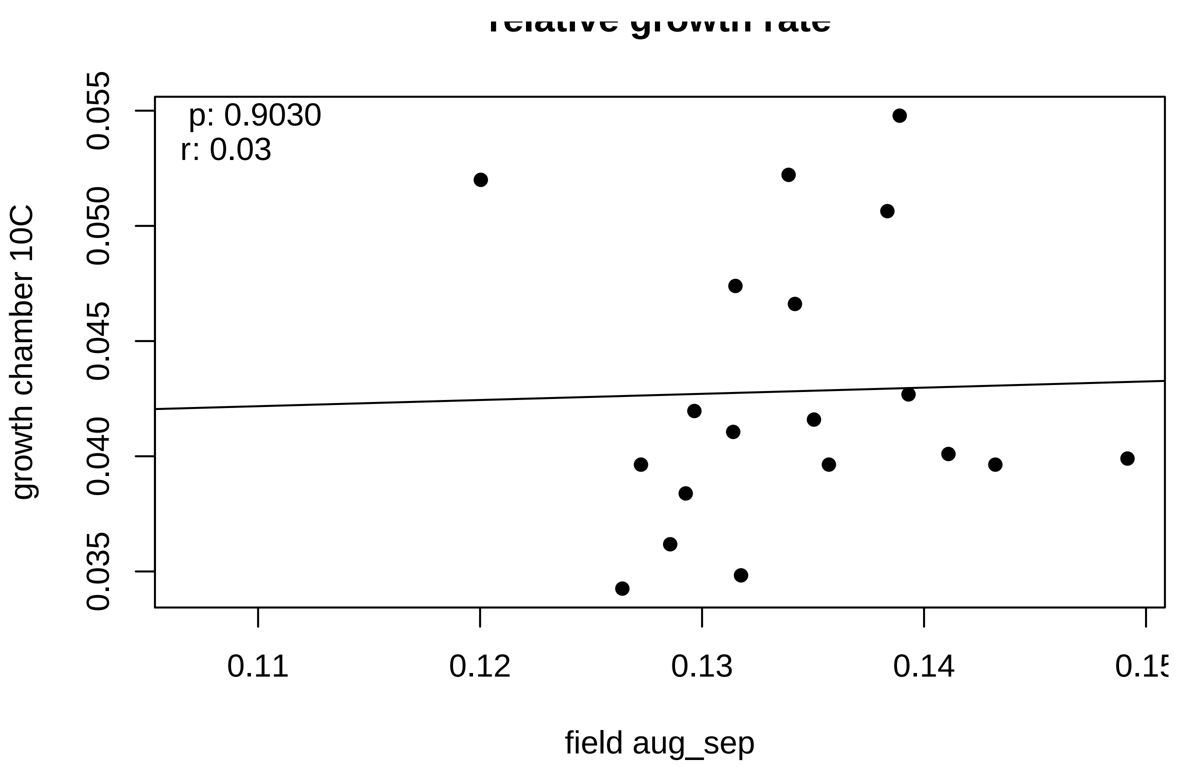


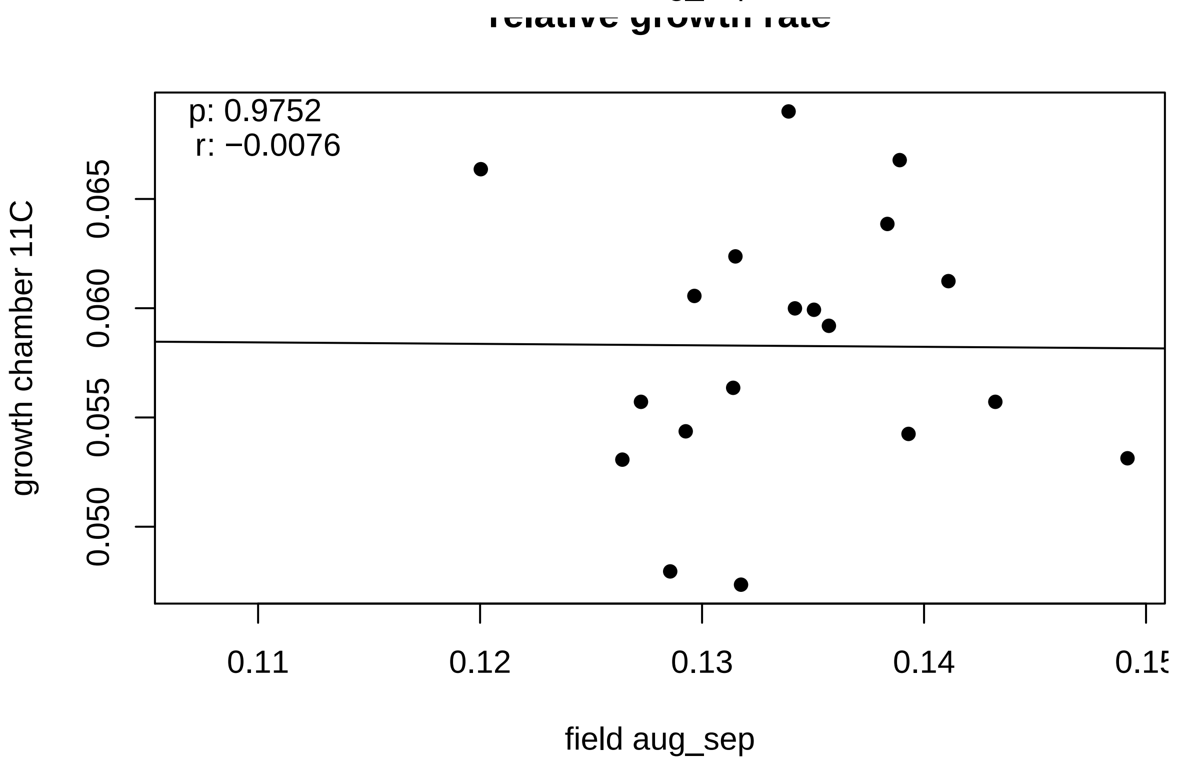


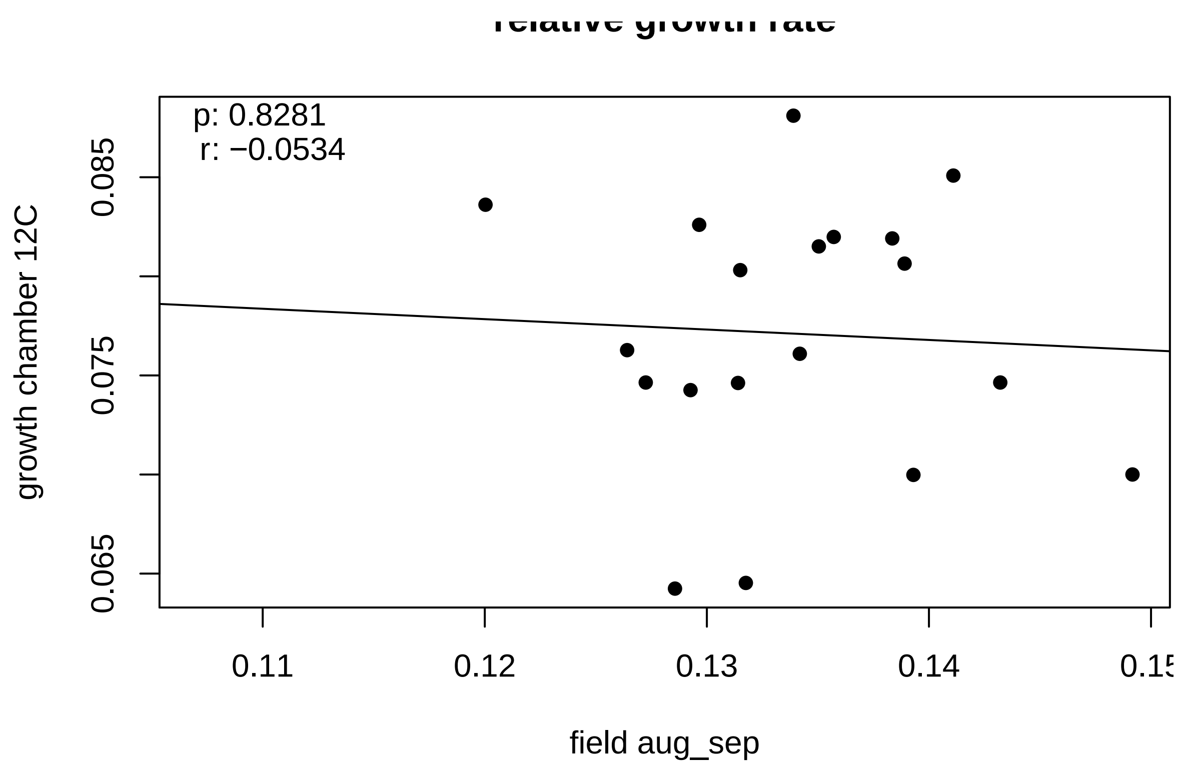
correlate growth chamber and field growth

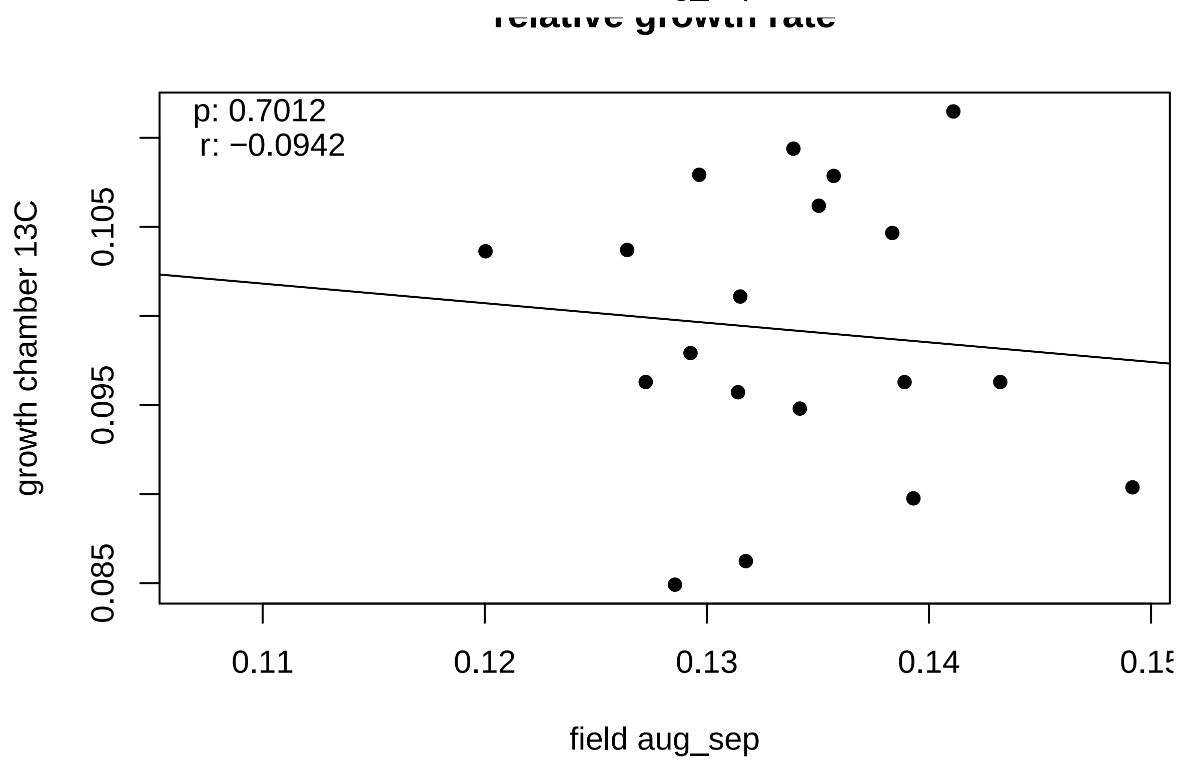
Correlate per temperature in growth chamber.

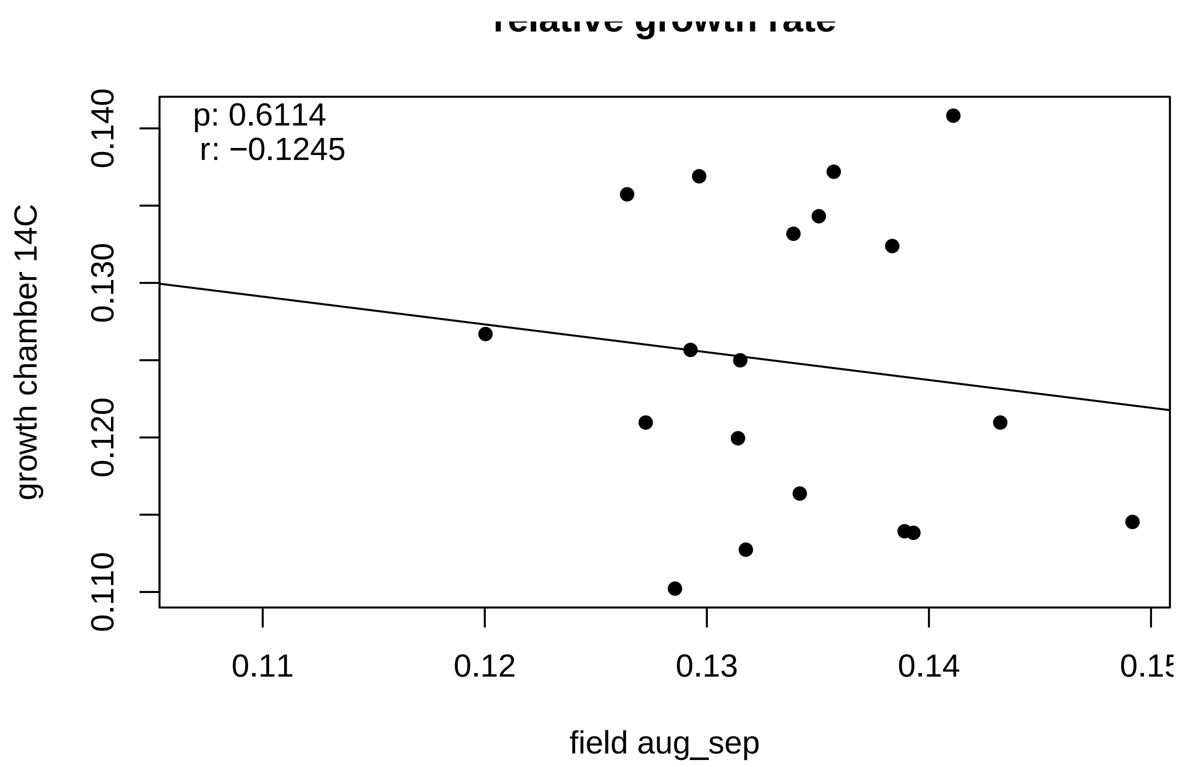
```
# merge results from RGR in growth chamber formn different temperatures to RGR in field
for(temp in names(RGR.GC))
{
  RGR.GC.C <- RGR.GC[[temp]]
  #RGR <- cbind(RGR, RGR.GC.C$relgr[match(RGR$acn, RGR.GC.C$acn)])
  RGR[, paste('rgr', temp, sep = '_')] <- RGR.GC.C$relgr[match(RGR$acn, RGR.GC.C$acn)]
}

for (period in unique(RGR$period))
{
  for (temp in names(RGR.GC))
  {
    RGR.p <- RGR[RGR$period == period, ]
    plot(RGR.p$rgr, RGR.p[,paste('rgr', temp, sep = '_')], xlab = paste('field', period), ylab = paste(
    abline(lm(RGR.p[,paste('rgr', temp, sep = '_')] ~ RGR.p$rgr))
    ct <- cor.test(RGR.p$rgr, RGR.p[,paste('rgr', temp, sep = '_')])
    text(x = par('usr')[1], y = par('usr')[4], paste('p: ', format.pval(ct$p.value, digits = 3, eps = 0
    text(x = par('usr')[1], y = par('usr')[4], paste('r: ', round(ct$estimate, 4), sep = '' ), adj = c(
  }

}
```

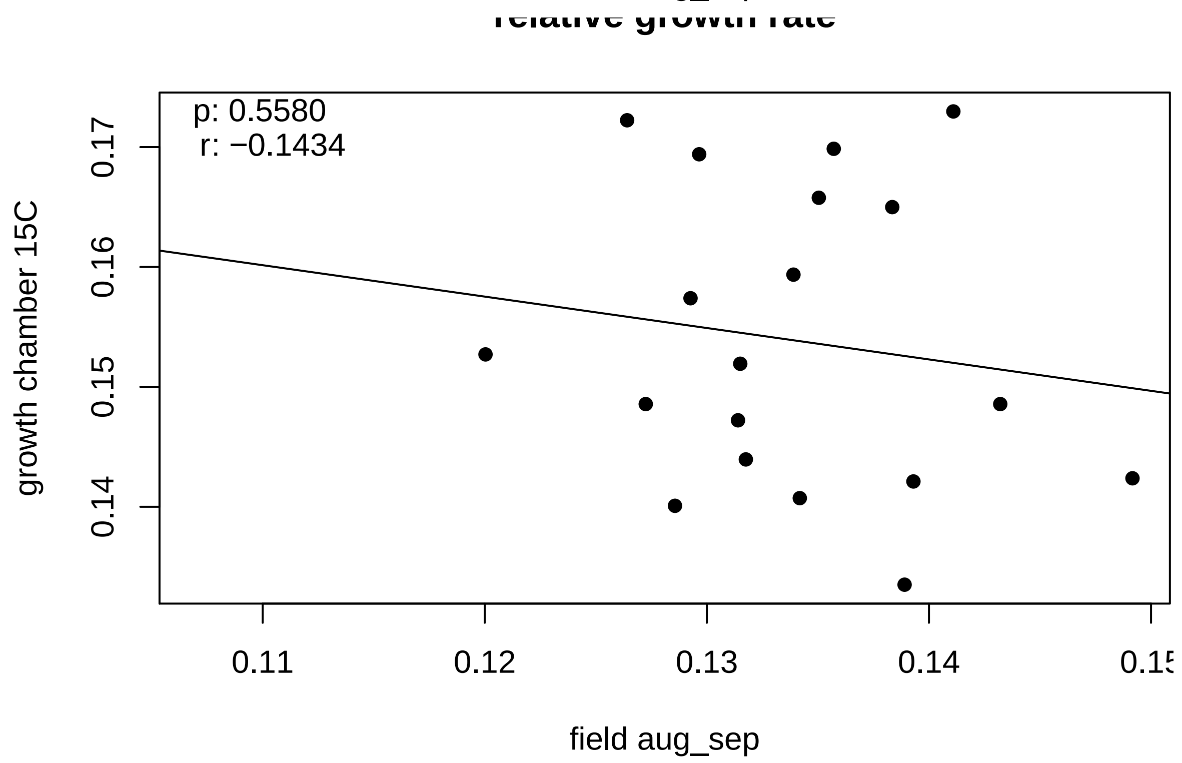

relative growth rate
p: 0.9030
r: 0.03
growth chamber 10C
0.035
0.040
0.045
0.050
0.055
0.11
0.12
0.13
0.14
0.15
field aug_sep

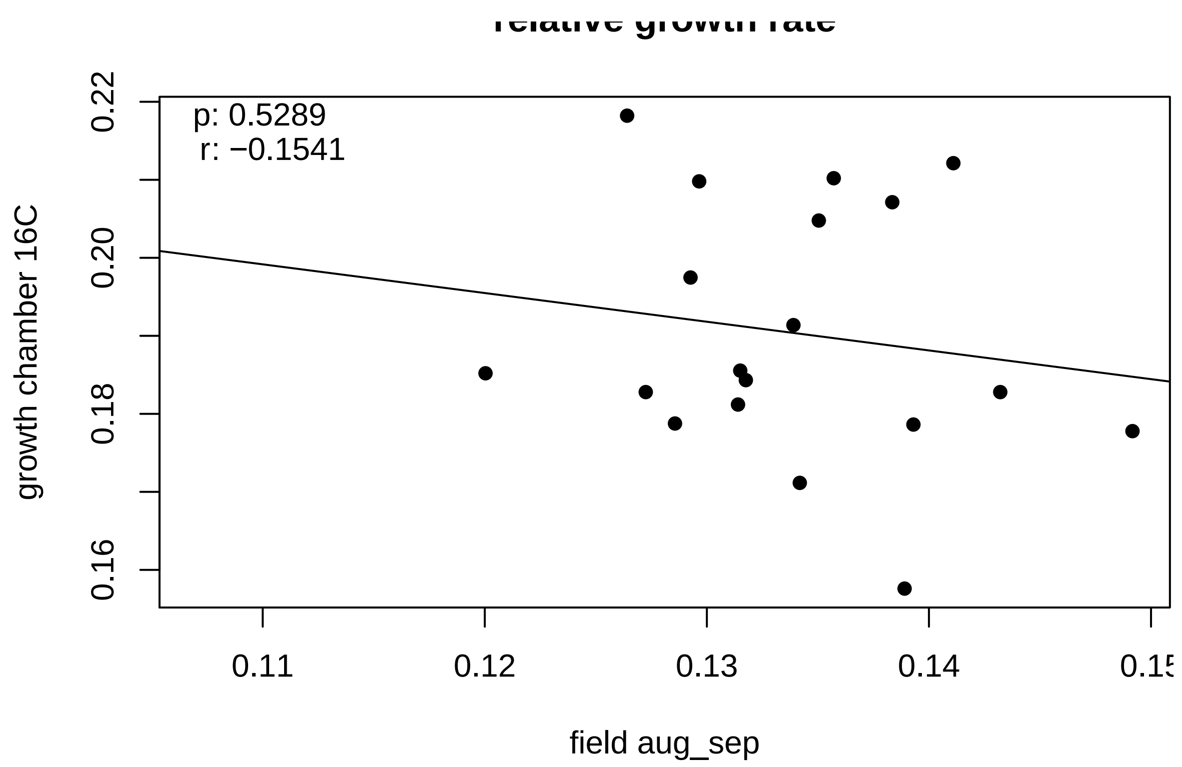

relative growth rate
p: 0.9752
r: −0.0076
growth chamber 11C
0.050
0.055
0.060
0.065
0.11
0.12
0.13
0.14
0.15
field aug_sep

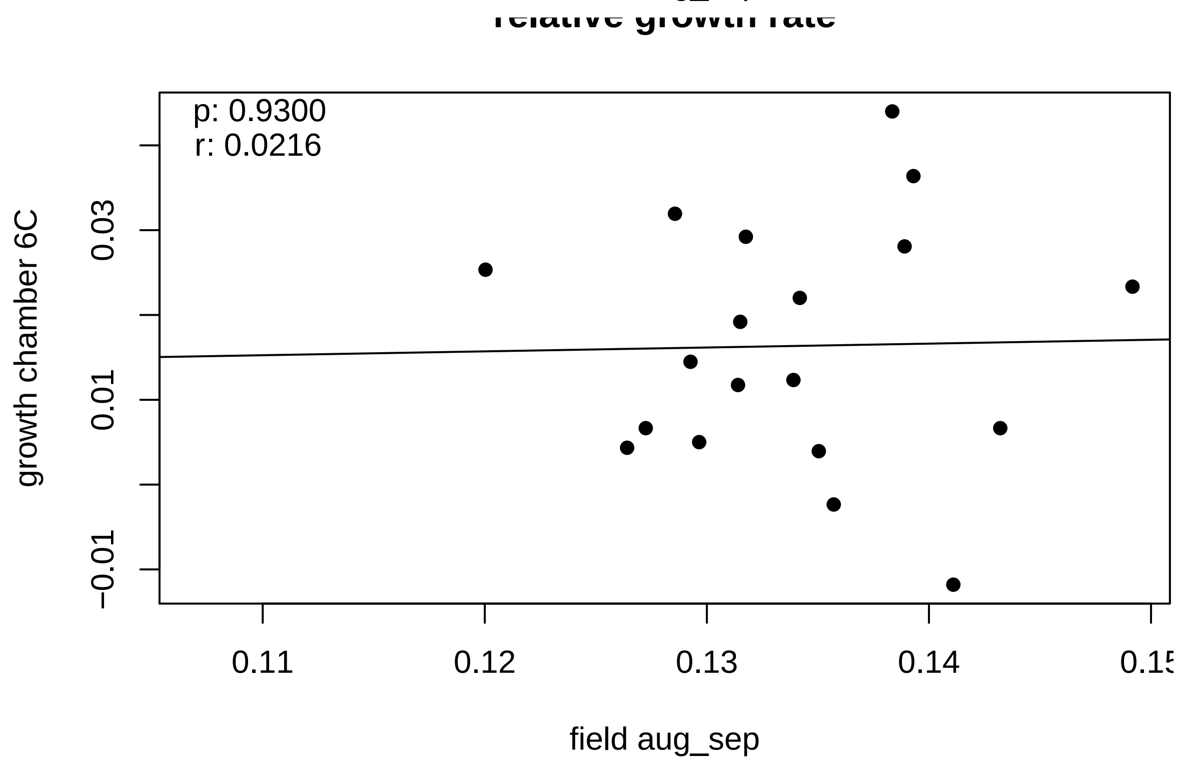

relative growth rate
p: 0.8281
r: −0.0534
growth chamber 12C
0.065
0.075
0.085
0.11
0.12
0.13
0.14
0.15
field aug_sep

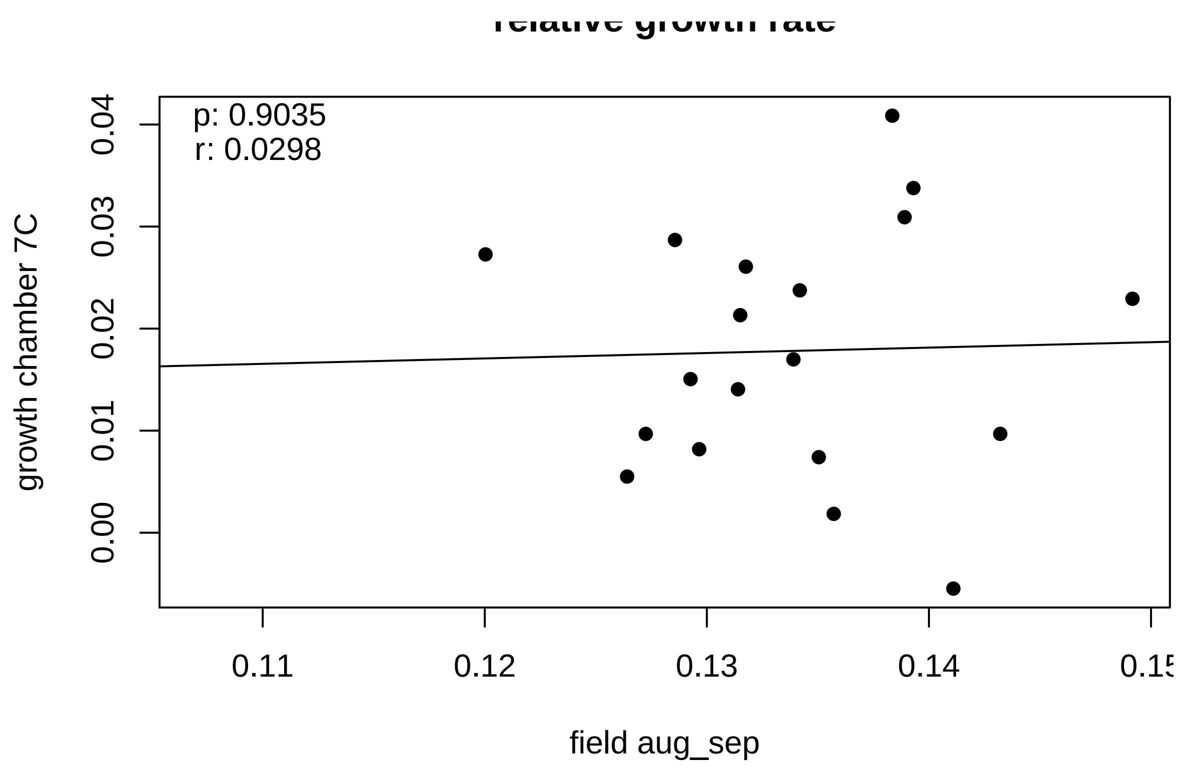

relative growth rate
p: 0.7012
r: −0.0942
growth chamber 13C
0.085
0.095
0.105
0.11
0.12
0.13
0.14
0.15
field aug_sep

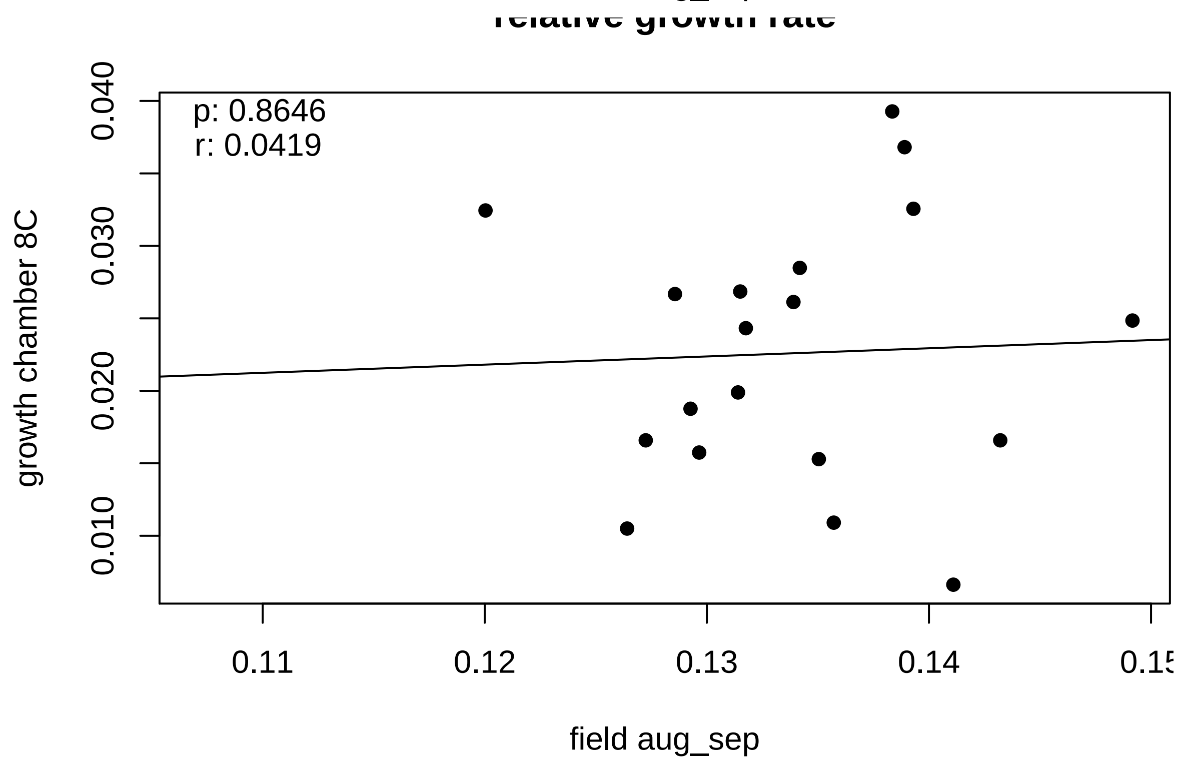

relative growth rate
p: 0.6114
r: −0.1245
growth chamber 14C
0.110
0.120
0.130
0.140
0.11
0.12
0.13
0.14
0.15
field aug_sep

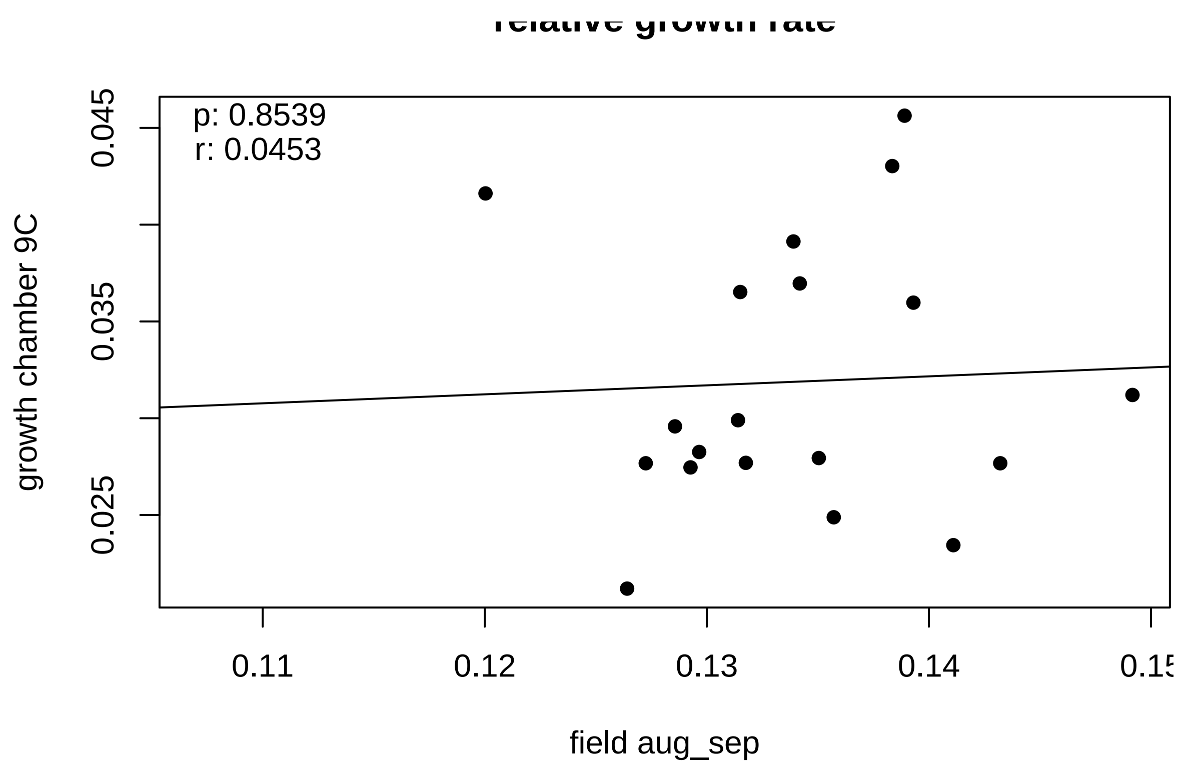

relative growth rate
p: 0.5580
r: −0.1434
growth chamber 15C
0.14
0.15
0.16
0.17
0.11
0.12
0.13
0.14
0.15
field aug_sep

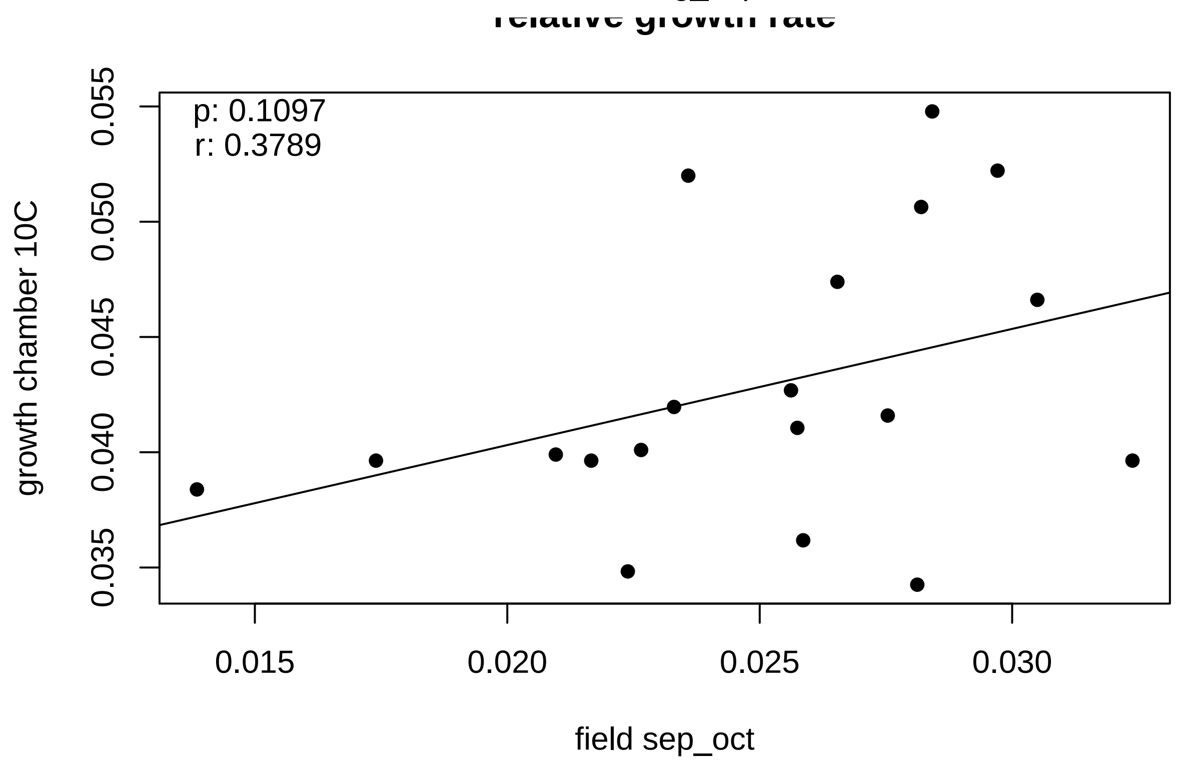

relative growth rate
p: 0.5289
r: −0.1541
growth chamber 16C
0.22
0.20
0.18
0.16
0.11
0.12
0.13
0.14
0.15
field aug_sep

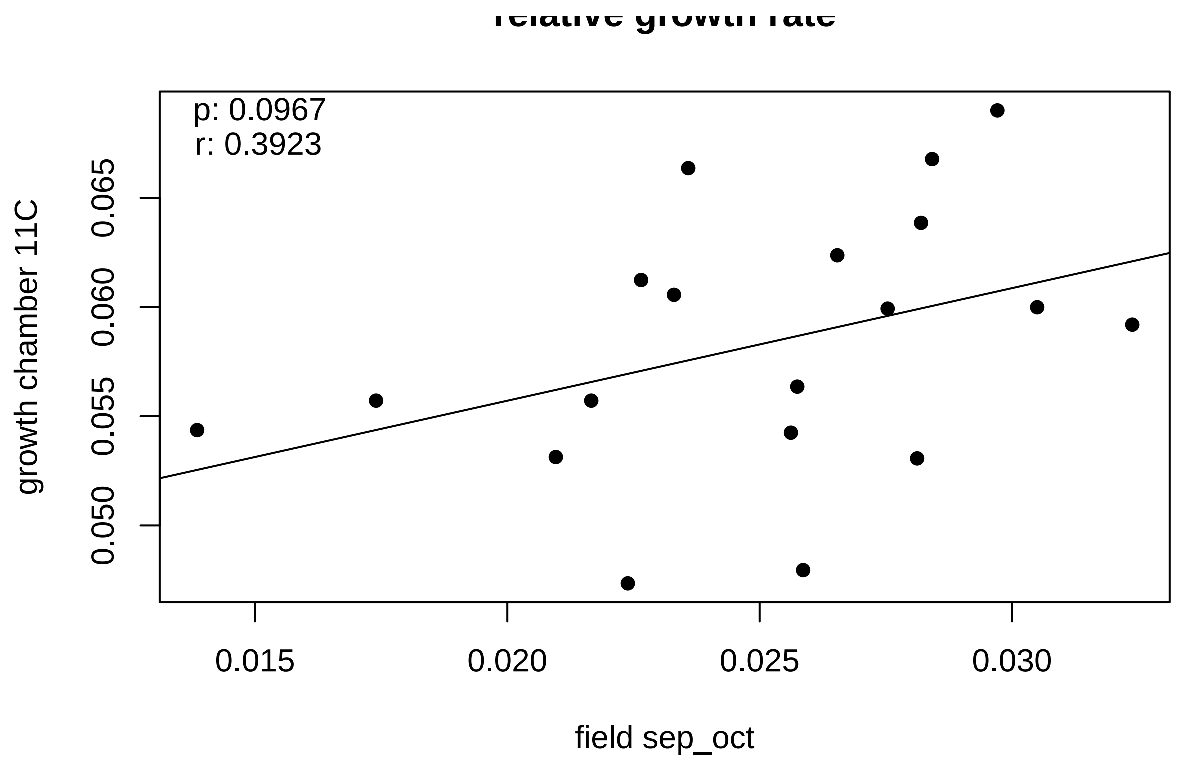

relative growth rate
p: 0.9300
r: 0.0216
growth chamber 6C
0.03
0.01
−0.01
0.11
0.12
0.13
0.14
0.15
field aug_sep

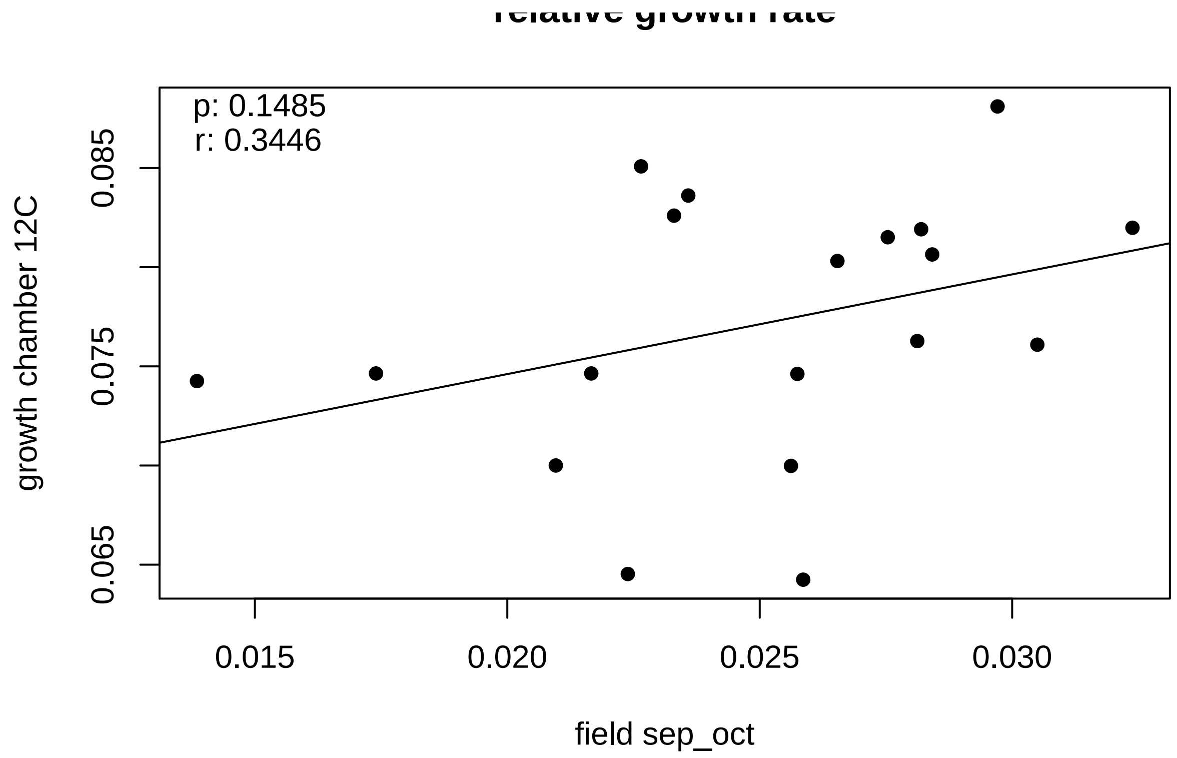

relative growth rate
p: 0.9035
r: 0.0298
growth chamber 7C
field aug_sep
0.00
0.01
0.02
0.03
0.04
0.11
0.12
0.13
0.14
0.15

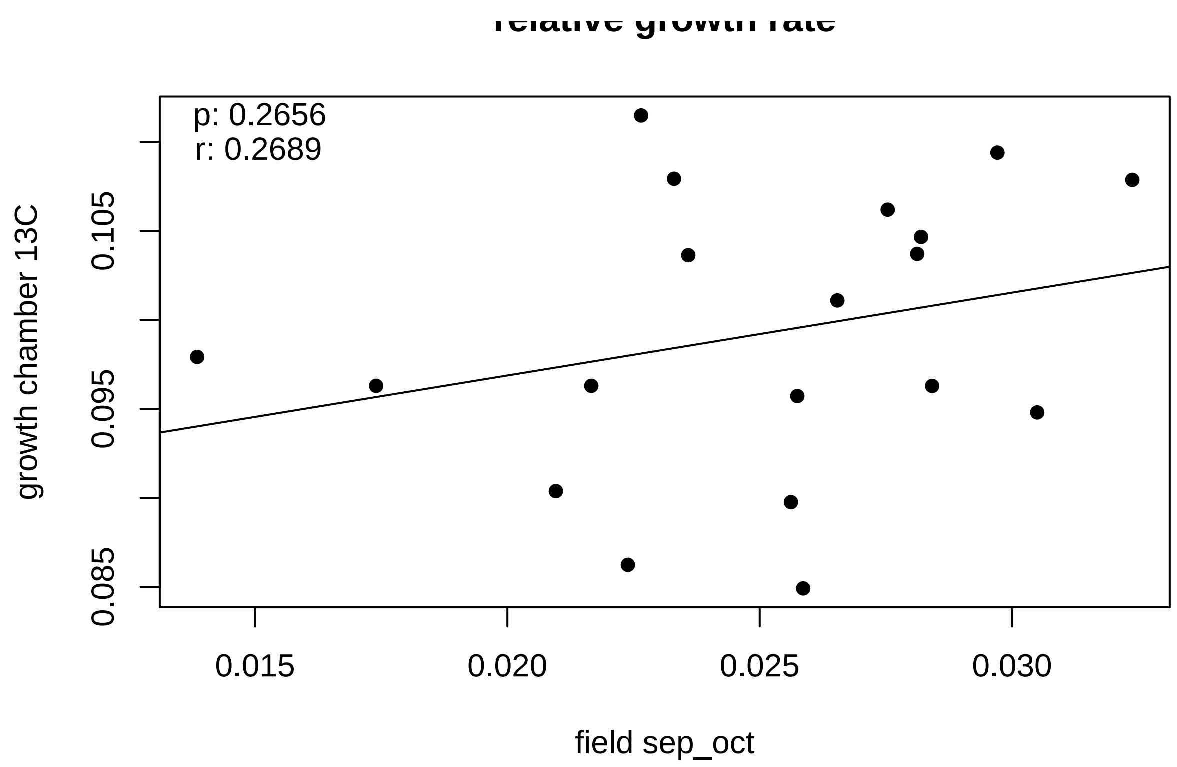

relative growth rate
p: 0.8646
r: 0.0419
growth chamber 8C
field aug_sep
0.010
0.020
0.030
0.040
0.11
0.12
0.13
0.14
0.15

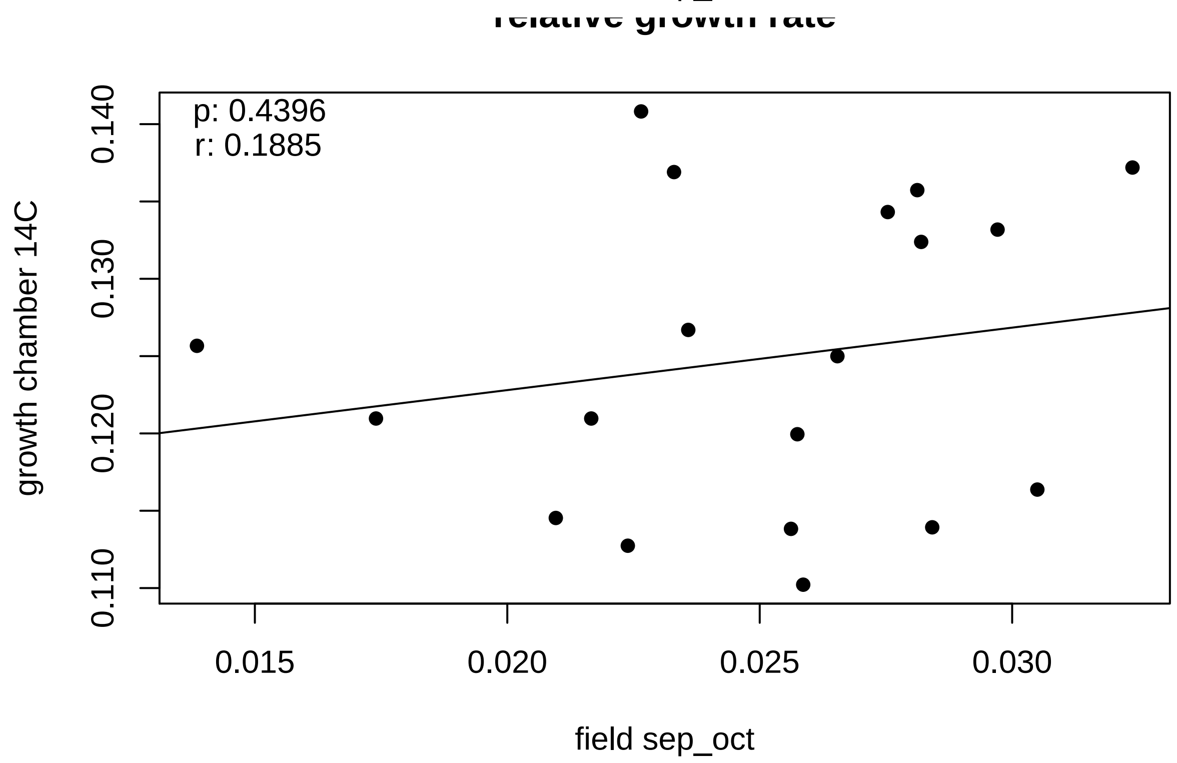

relative growth rate
p: 0.8539
r: 0.0453
growth chamber 9C
0.025
0.035
0.045
0.11
0.12
0.13
0.14
0.15
field aug_sep

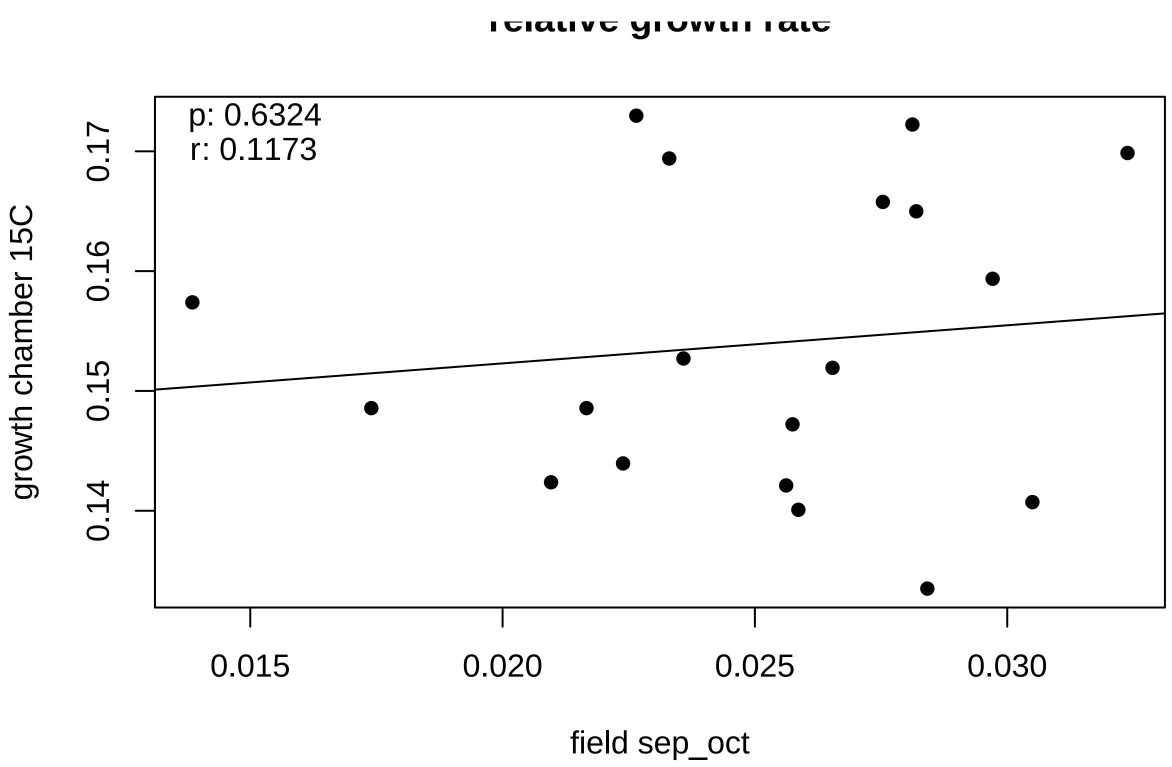

relative growth rate
p: 0.1097
r: 0.3789
growth chamber 10C
0.035
0.040
0.045
0.050
0.055
0.015
0.020
0.025
0.030
field sep_oct

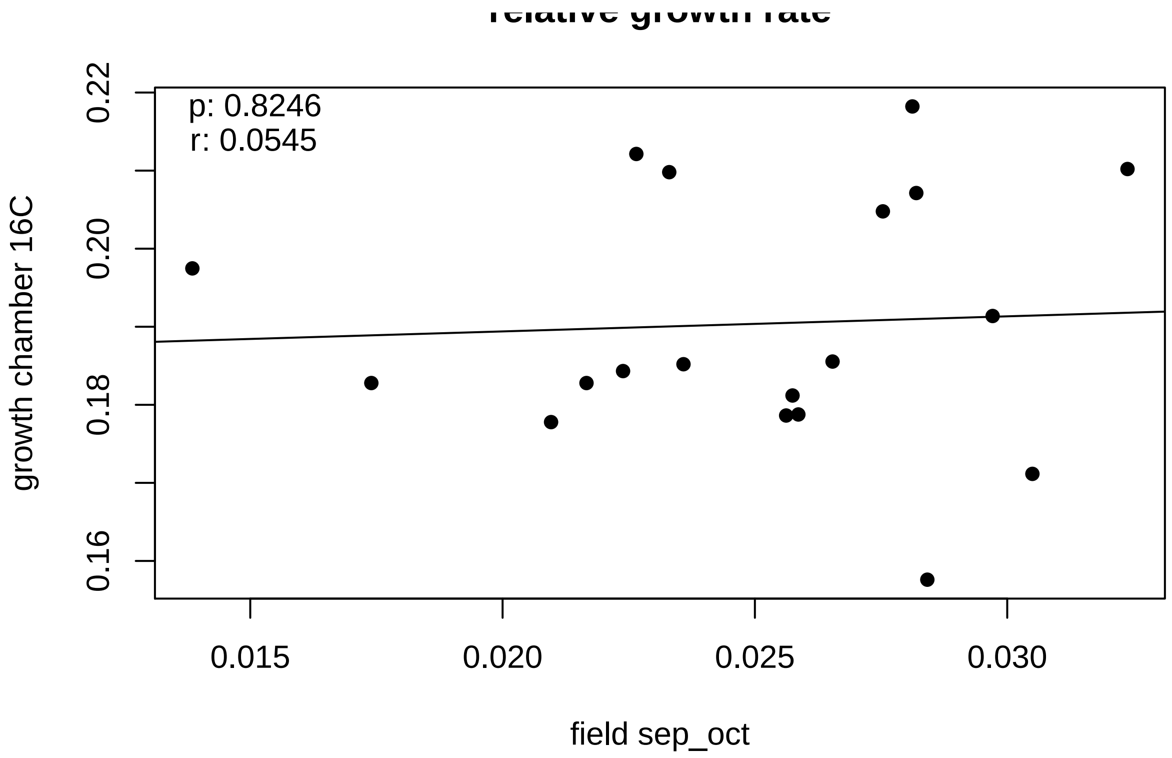

relative growth rate
p: 0.0967
r: 0.3923
growth chamber 11C
0.050
0.055
0.060
0.065
0.015
0.020
0.025
0.030
field sep_oct

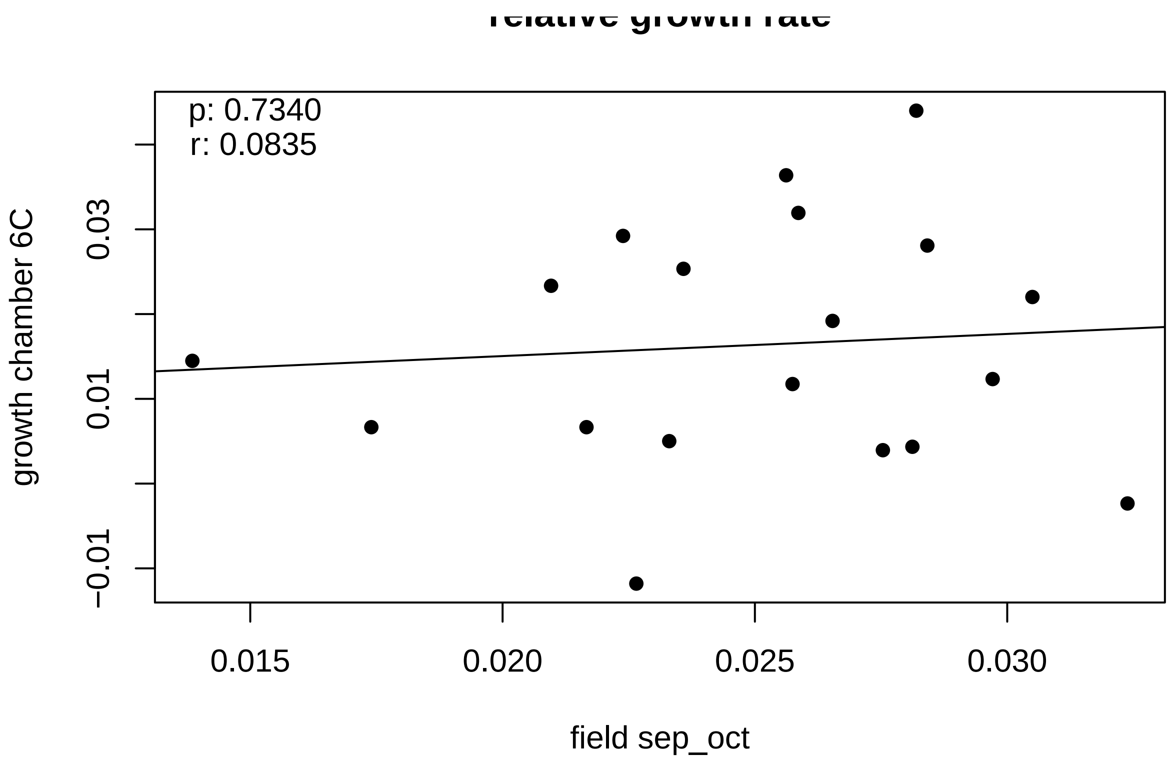

relative growth rate
p: 0.1485
r: 0.3446
growth chamber 12C
0.065
0.075
0.085
0.015
0.020
0.025
0.030
field sep_oct

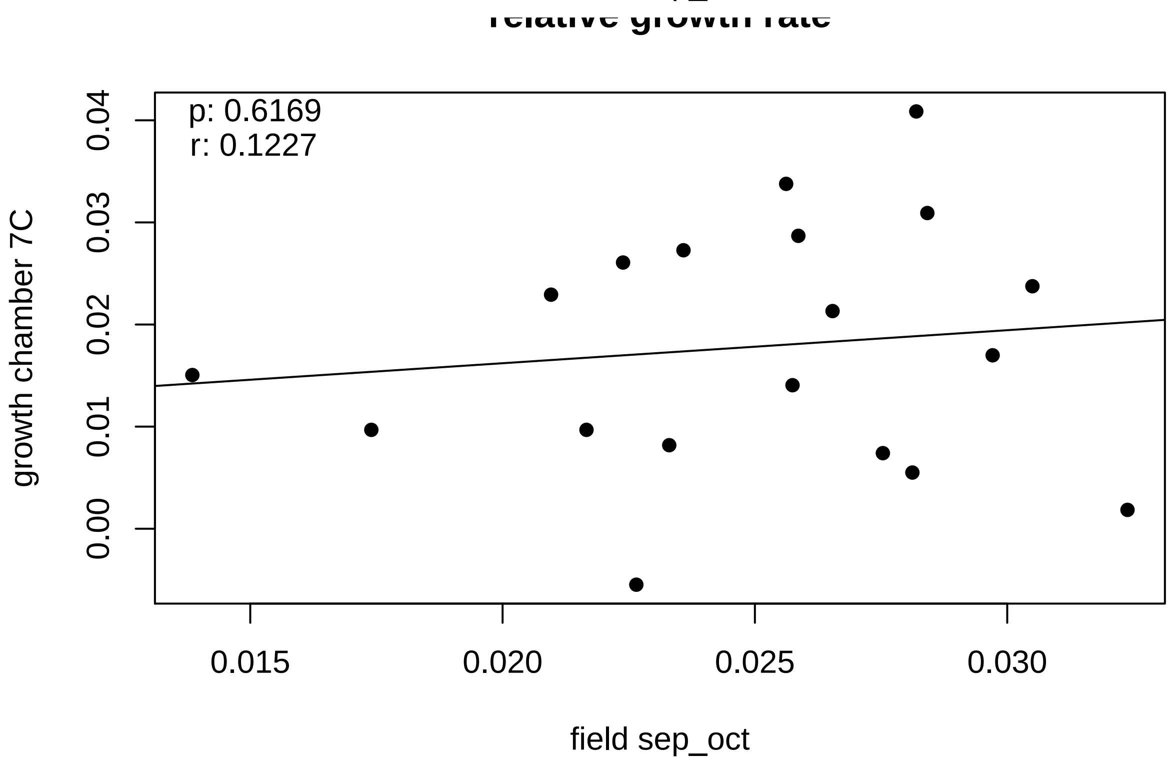

relative growth rate
p: 0.2656
r: 0.2689
growth chamber 13C
0.085
0.095
0.105
0.015
0.020
0.025
0.030
field sep_oct

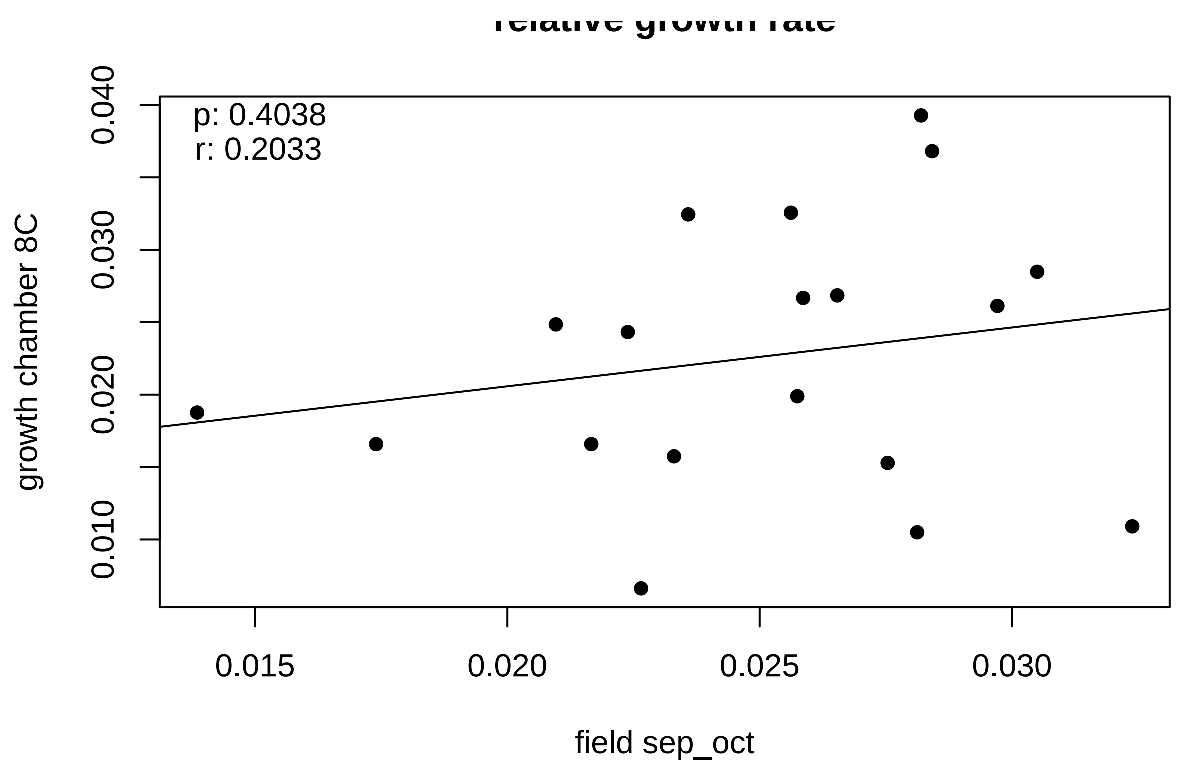

relative growth rate
p: 0.4396
r: 0.1885
growth chamber 14C
0.110
0.120
0.130
0.140
0.015
0.020
0.025
0.030
field sep_oct

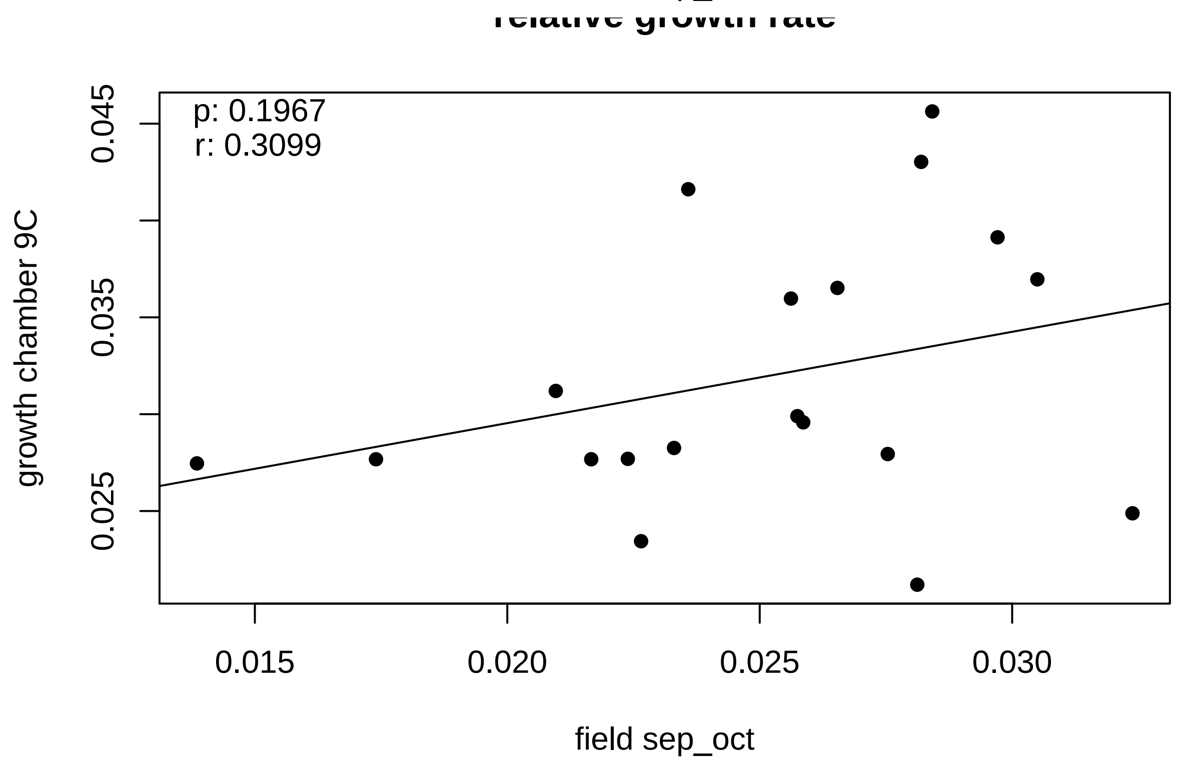

relative growth rate
p: 0.6324
r: 0.1173
growth chamber 15C
0.14
0.15
0.16
0.17
0.015
0.020
0.025
0.030
field sep_oct


relative growth rate
p: 0.8246
r: 0.0545
growth chamber 16C
0.16
0.18
0.20
0.22
0.015
0.020
0.025
0.030
field sep_oct


relative growth rate
p: 0.7340
r: 0.0835
growth chamber 6C
field sep_oct
0.015
0.020
0.025
0.030
−0.01
0.01
0.03


relative growth rate
p: 0.6169
r: 0.1227
growth chamber 7C
field sep_oct
0.015
0.020
0.025
0.030
0.00
0.01
0.02
0.03
0.04


relative growth rate
p: 0.4038
r: 0.2033
growth chamber 8C
field sep_oct
0.010
0.020
0.030
0.040
0.015
0.020
0.025
0.030


relative growth rate
p: 0.1967
r: 0.3099
growth chamber 9C
field sep_oct
0.025
0.035
0.045
0.015
0.020
0.025
0.030