Soy Beans

Heba+Valentine

2024-09-10

To install the needed packages if it is not already installed This need to be run only the first time you use this package

# install.packages("dplyr")  
# install.packages("ggplot2")  
# install.packages("devtools")  
# library(devtools)  
# install\_github("DoseResponse/drc",dependencies=TRUE)  
# install\_github("DoseResponse/bmd",dependencies=TRUE)

To load all the packages needed for this code this should be loaded each time you open R to run this code

library(drc)

## Loading required package: MASS

## Loading required package: drcData

##   
## 'drc' has been loaded.

## Please cite R and 'drc' if used for a publication,

## for references type 'citation()' and 'citation('drc')'.

##   
## Attaching package: 'drc'

## The following objects are masked from 'package:stats':  
##   
## gaussian, getInitial

library(bmd)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

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

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

library(ggplot2)

Reading the data set: the file is called “PE2\_germination\_data.csv” Note: if the numbers are: - in the american system use “read.csv(file.choose())” - in the european system use “read.csv2(file.choose())”

m <- read.csv(file.choose())  
head(m)

## experiment rep treatment start end obs  
## 1 1 R1 21 0 1 0  
## 2 1 R1 21 1 2 1  
## 3 1 R1 21 2 3 2  
## 4 1 R1 21 3 4 14  
## 5 1 R1 21 4 5 12  
## 6 1 R1 21 5 6 7

tail(m)

## experiment rep treatment start end obs  
## 203 2 R2 2.5 7 8 3  
## 204 2 R2 2.5 8 9 1  
## 205 2 R2 2.5 9 10 2  
## 206 2 R2 2.5 10 11 3  
## 207 2 R2 2.5 11 12 2  
## 208 2 R2 2.5 12 Inf 22

#exploratory graphs We plot the raw data to get an idea of what the data looks like

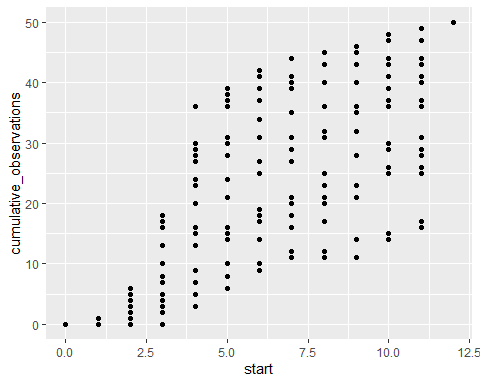
#calculating cumulative observations  
m\_cumu <- m %>% #we create a new dataframe with an extra column for cumulative observations  
 group\_by(experiment, rep, treatment) %>% #cumulative observations are grouped by factors (here we have three factors, this can be adjusted for more or less factors)  
 mutate(cumulative\_observations = cumsum(obs)) #cumulative observations are stored in a new column  
  
head(m\_cumu)

## # A tibble: 6 × 7  
## # Groups: experiment, rep, treatment [1]  
## experiment rep treatment start end obs cumulative\_observations  
## <int> <chr> <dbl> <int> <dbl> <int> <int>  
## 1 1 R1 21 0 1 0 0  
## 2 1 R1 21 1 2 1 1  
## 3 1 R1 21 2 3 2 3  
## 4 1 R1 21 3 4 14 17  
## 5 1 R1 21 4 5 12 29  
## 6 1 R1 21 5 6 7 36

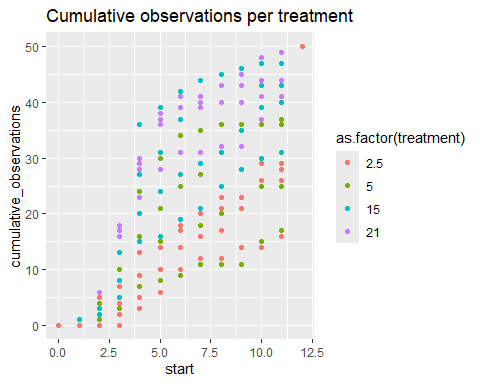
tail(m\_cumu)

## # A tibble: 6 × 7  
## # Groups: experiment, rep, treatment [1]  
## experiment rep treatment start end obs cumulative\_observations  
## <int> <chr> <dbl> <int> <dbl> <int> <int>  
## 1 2 R2 2.5 7 8 3 20  
## 2 2 R2 2.5 8 9 1 21  
## 3 2 R2 2.5 9 10 2 23  
## 4 2 R2 2.5 10 11 3 26  
## 5 2 R2 2.5 11 12 2 28  
## 6 2 R2 2.5 12 Inf 22 50

#plot cumulative observations  
ggplot(m\_cumu, aes(x = start, y = cumulative\_observations)) +  
 geom\_point()



#plot cumulative observations, grouped by treatment  
ggplot(m\_cumu, aes(x = start, y = cumulative\_observations, color = as.factor(treatment))) +  
 geom\_point()+  
 ggtitle("Cumulative observations per treatment")



#model fits

##dose-response model (drm) fitting

if we need to get help on the specific function that we are using we can do this:

?drm

## starting httpd help server ... done

getMeanFunctions()

## Log-logistic (ED50 as parameter) with lower limit at 0 and upper limit at 1   
## (2 parameters)   
## In 'drc': LL.2   
##   
## Log-logistic (ED50 as parameter) with lower limit at 0   
## (3 parameters)   
## In 'drc': LL.3   
##   
## Log-logistic (ED50 as parameter) with upper limit at 1   
## (3 parameters)   
## In 'drc': LL.3u   
##   
## Log-logistic (ED50 as parameter)   
## (4 parameters)   
## In 'drc': LL.4   
##   
## Generalized log-logistic (ED50 as parameter)   
## (5 parameters)   
## In 'drc': LL.5   
##   
## Weibull (type 1) with lower limit at 0 and upper limit at 1   
## (2 parameters)   
## In 'drc': W1.2   
##   
## Weibull (type 1) with lower limit at 0   
## (3 parameters)   
## In 'drc': W1.3   
##   
## Weibull (type 1)   
## (4 parameters)   
## In 'drc': W1.4   
##   
## Weibull (type 2) with lower limit at 0 and upper limit at 1   
## (2 parameters)   
## In 'drc': W2.2   
##   
## Weibull (type 2) with lower limit at 0   
## (3 parameters)   
## In 'drc': W2.3   
##   
## Weibull (type 2)   
## (4 parameters)   
## In 'drc': W2.4   
##   
## Brain-Cousens (hormesis) with lower limit fixed at 0   
## (4 parameters)   
## In 'drc': BC.4   
##   
## Brain-Cousens (hormesis)   
## (5 parameters)   
## In 'drc': BC.5   
##   
## Log-logistic (log(ED50) as parameter) with lower limit at 0 and upper limit at 1   
## (2 parameters)   
## In 'drc': LL2.2   
##   
## Log-logistic (log(ED50) as parameter) with lower limit at 0   
## (3 parameters)   
## In 'drc': LL2.3   
##   
## Log-logistic (log(ED50) as parameter) with upper limit at 1   
## (3 parameters)   
## In 'drc': LL2.3u   
##   
## Log-logistic (log(ED50) as parameter)   
## (4 parameters)   
## In 'drc': LL2.4   
##   
## Generalised log-logistic (log(ED50) as parameter)   
## (5 parameters)   
## In 'drc': LL2.5   
##   
## Asymptotic regression with lower limit at 0   
## (2 parameters)   
## In 'drc': AR.2   
##   
## Shifted asymptotic regression   
## (3 parameters)   
## In 'drc': AR.3   
##   
## Michaelis-Menten   
## (2 parameters)   
## In 'drc': MM.2   
##   
## Shifted Michaelis-Menten   
## (3 parameters)   
## In 'drc': MM.3

To Fit the model using the drm function:

model\_LL<-drm(obs~start+end, # The formula (in terms of the variables in the data)  
 type="event", # the type of the data  
 fct=LL.3(), # the function of the model   
 data= m, # specifying the name of the data set  
 curveid = treatment, # the curve identifier (grouping variable)  
 separate = TRUE) # to fit every treatment independently

## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]

we fit the other two possible models:

model\_W1<-drm(obs~start+end,type="event", fct=W1.3(),data= m, curveid = treatment, separate = TRUE)

## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]

model\_W2<-drm(obs~start+end,type="event", fct=W2.3(),data= m, curveid = treatment, separate = TRUE)

## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]  
## Warning in matrix(c(predVec, resVec), length(dose), 2): data length differs  
## from size of matrix: [96 != 12 x 2]

##model selection

We select the best model fit per treatment using Akaike’s Information Criterion: the smaller the AIC, the better the fit.

?AIC()

For each treatment and each model we calculate the AIC

aic\_df\_LL <- data.frame(  
 Treatment = names(lapply(model\_LL$objList, function(x) AIC(x))),  
 AIC\_LL3 = unlist(lapply(model\_LL$objList, function(x) AIC(x)))  
)  
  
aic\_df\_W1 <- data.frame(  
 Treatment = names(lapply(model\_W1$objList, function(x) AIC(x))),  
 AIC\_W1 = unlist(lapply(model\_W1$objList, function(x) AIC(x)))  
)  
  
aic\_df\_W2 <- data.frame(  
 Treatment = names(lapply(model\_W2$objList, function(x) AIC(x))),  
 AIC\_W2 = unlist(lapply(model\_W2$objList, function(x) AIC(x)))  
)

We combine all AICs in one dataframe to ease comparison

AIC\_df <- data.frame(  
 Treatment = names(lapply(model\_LL$objList, function(x) AIC(x))),  
 AIC\_LL = unlist(lapply(model\_LL$objList, function(x) AIC(x))),  
 AIC\_W1 = unlist(lapply(model\_W1$objList, function(x) AIC(x))),  
 AIC\_W2 = unlist(lapply(model\_W2$objList, function(x) AIC(x)))  
)  
AIC\_df$MinAIC <-   
 apply(AIC\_df[, 2:4], 1, function(x) {colnames(AIC\_df)[2:4][which.min(x)]}) #we create a column that displays the name of the model with the lowest AIC  
  
rownames(AIC\_df) <- NULL  
  
print(AIC\_df)

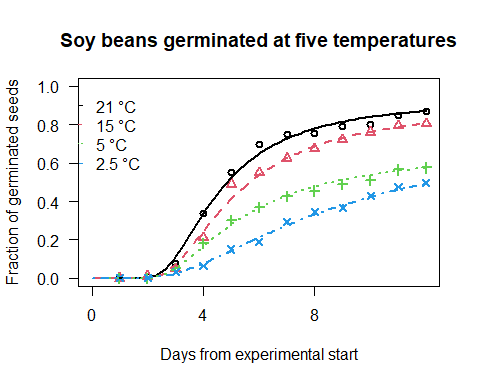
## Treatment AIC\_LL AIC\_W1 AIC\_W2 MinAIC  
## 1 21 956.8821 946.2194 1005.3958 AIC\_W1  
## 2 15 970.0964 964.8335 996.8095 AIC\_W1  
## 3 5 879.1615 864.8927 893.5418 AIC\_W1  
## 4 2.5 826.7805 824.7821 829.0154 AIC\_W1

–> since the minimum AIC across all the treatments is for W1, this is the model that we are going to continue the analysis with

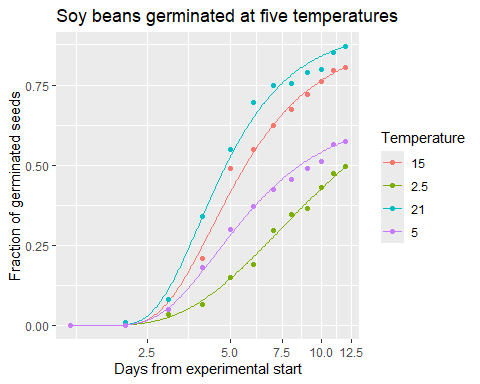
#treatment comparison

##parameters

plot(model\_W1, log="", ylim=c(0,1), xlim=c(0,12), col=TRUE,lwd ="2",  
 legendPos=c(2, 1.0), legendText = c("21 °C","15 °C", "5 °C","2.5 °C"),  
 xlab="Days from experimental start", ylab="Fraction of germinated seeds",  
 main= "Soy beans germinated at five temperatures")



qplotDrc(model\_W1,  
 col=TRUE,  
 xlab="Days from experimental start", ylab="Fraction of germinated seeds") +  
 ggtitle("Soy beans germinated at five temperatures")+  
 labs(col = "Temperature")



##confidence intervals

model\_W1$objList #parameters for each treatment

## $`21`  
##   
## A 'drc' model.  
##   
## Call:  
## drm(formula = y ~ x, curveid = curveid, weights = weights, subset = curveid == uniCur[i], fct = fct, type = type, control = control, separate = FALSE)  
##   
## Coefficients:  
## b:(Intercept) d:(Intercept) e:(Intercept)   
## -2.5833 0.9229 4.0139   
##   
##   
## $`15`  
##   
## A 'drc' model.  
##   
## Call:  
## drm(formula = y ~ x, curveid = curveid, weights = weights, subset = curveid == uniCur[i], fct = fct, type = type, control = control, separate = FALSE)  
##   
## Coefficients:  
## b:(Intercept) d:(Intercept) e:(Intercept)   
## -2.3186 0.8905 4.4632   
##   
##   
## $`5`  
##   
## A 'drc' model.  
##   
## Call:  
## drm(formula = y ~ x, curveid = curveid, weights = weights, subset = curveid == uniCur[i], fct = fct, type = type, control = control, separate = FALSE)  
##   
## Coefficients:  
## b:(Intercept) d:(Intercept) e:(Intercept)   
## -2.1016 0.6589 4.6463   
##   
##   
## $`2.5`  
##   
## A 'drc' model.  
##   
## Call:  
## drm(formula = y ~ x, curveid = curveid, weights = weights, subset = curveid == uniCur[i], fct = fct, type = type, control = control, separate = FALSE)  
##   
## Coefficients:  
## b:(Intercept) d:(Intercept) e:(Intercept)   
## -1.3537 0.8506 7.6258

lapply(model\_W1$objList, confint) #confidence interval for each parameter for each treatment

## $`21`  
## 2.5 % 97.5 %  
## b:(Intercept) -2.997216 -2.1694591  
## d:(Intercept) 0.864864 0.9809946  
## e:(Intercept) 3.738541 4.2891989  
##   
## $`15`  
## 2.5 % 97.5 %  
## b:(Intercept) -2.7312996 -1.905990  
## d:(Intercept) 0.8123935 0.968614  
## e:(Intercept) 4.0836206 4.842702  
##   
## $`5`  
## 2.5 % 97.5 %  
## b:(Intercept) -2.6076735 -1.5954470  
## d:(Intercept) 0.5597144 0.7580087  
## e:(Intercept) 4.0855441 5.2069987  
##   
## $`2.5`  
## 2.5 % 97.5 %  
## b:(Intercept) -1.8616105 -0.8458383  
## d:(Intercept) 0.4972011 1.2039199  
## e:(Intercept) 4.7780556 10.4735194

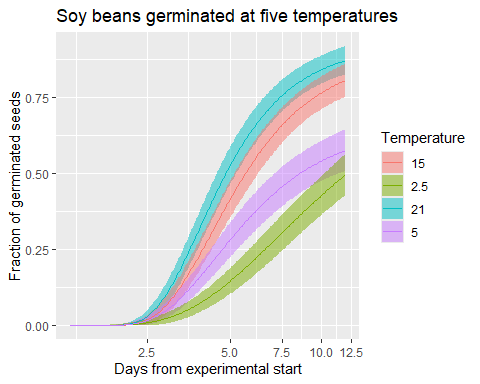
for treatment 21

the coefficients’ estimates are b:(Intercept) d:(Intercept) e:(Intercept)  
-2.5833 0.9229 4.0139 so, the upper limit (d) is 0.9229 the inflection point (e) is 4.0139 the slope at the inflection point (b) is -2.5833

and the confidence interval 2.5 % is the lower limit and 97.5 % is the upper limit of a 95% confidence interval for the mentioned estimate.

To add the confidence bands around the curves to be able to compare them we add: type=“confidence”

qplotDrc(model\_W1, type="confidence", col=T,  
 xlab="Days from experimental start", ylab="Fraction of germinated seeds") +  
 ggtitle("Soy beans germinated at five temperatures")+  
 labs(fill = "Temperature", col = "Temperature")



##effective dose 50

lapply(model\_W1$objList, function(x) ED(x, 50, display=FALSE))

## $`21`  
## Estimate Std. Error  
## e:21:50 4.625718 0.1928964  
##   
## $`15`  
## Estimate Std. Error  
## e:15:50 5.227481 0.2770512  
##   
## $`5`  
## Estimate Std. Error  
## e:5:50 5.531535 0.4282354  
##   
## $`2.5`  
## Estimate Std. Error  
## e:2.5:50 9.996947 2.388871