Oil palm SOC and leaf litter decomposition

Marcus Schmidt

8 November 2019

### Background

In the following course, we are looking at two processes in oil palm plantations: 1) Soil organic carbon (SOC) remaining after deforestation and 2) Decomposition of oil palm leafs

### Part 1 - SOC remaining

This data is part of a study on deforestation and land luse conversion in Indonesia, Cameroon and Peru. You can find the whole paper here: <https://www.pnas.org/content/112/32/9956>

Let’s fist load our library. It is the tidyverse library, which includes useful tools for data handling and visualization, i.e. the ggplot 2 package.

library("tidyverse")

Let’s now read in our data and look at the upper part:

DATA=read.csv("data/van-straaten-soc.csv",h=T)  
head(DATA)

I like to shorten our variables a little bit, it will be easier to type then:

DATA$t<-DATA$Time.since.Deforestation  
DATA$soc\_pc<-DATA$SOC\_prop\_top10\_C2

Next, I apply a filter to exclude observations that don’t state the country. We also want only oil palm observations and lastly, clay content shouldn’t change more than 20%.

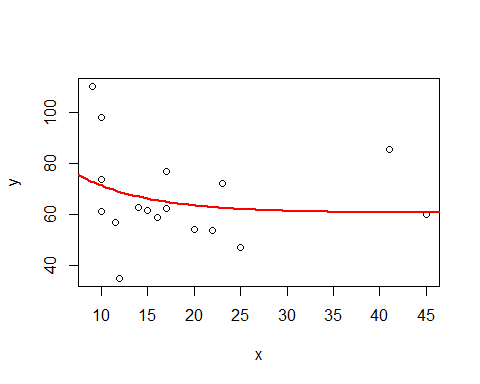
DATA<-DATA%>%filter(Country!="")%>%filter(Landuse == "Oil palm")  
DATA<-DATA%>%filter(abs(d\_Clay\_50to100)<20)

Now, for some of the following coding, it’s easier to give a shorter name to some variables. We choose x and y because this we will have on our axes:

y <- DATA$soc\_pc #  
x <- DATA$t

Let’s plot this in a very basic form, do the fitting and then add the line:

xx <- seq(0,605, length=1500)  
plot(y~x)  
  
fit2=nls(y~a+(100-a)\*exp(-k\*x),start=list(a=60, k=0.15), algorithm="port")  
  
lines(xx, predict(fit2, data.frame(x=xx)), col="red", lwd=2, lty=1)



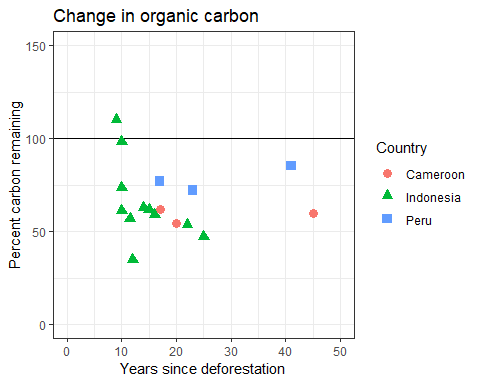
summary(fit2)

##   
## Formula: y ~ a + (100 - a) \* exp(-k \* x)  
##   
## Parameters:  
## Estimate Std. Error t value Pr(>|t|)   
## a 60.4117 9.9786 6.054 2.21e-05 \*\*\*  
## k 0.1296 0.1066 1.216 0.243   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 17.99 on 15 degrees of freedom  
##   
## Algorithm "port", convergence message: relative convergence (4)  
## (4 observations deleted due to missingness)

The plot will look much nicer in ggplot2, however:

g2 <- ggplot(DATA,aes(t,soc\_pc, shape = Country, colour = Country)) +  
 geom\_hline(yintercept=100) +  
 xlim(0,50) +  
 ylim(0,150) +  
 geom\_point(show.legend = T, size = 3) +  
 ggtitle("Change in organic carbon") +  
 xlab("Years since deforestation") +  
 ylab("Percent carbon remaining") +  
 theme\_bw()  
g2

## Warning: Removed 4 rows containing missing values (geom\_point).

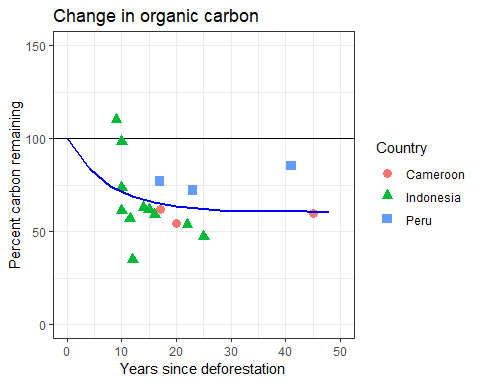


Finally, we will add the curve to our ggplot:

func3<- function(x) {60.4+(100-60.4)\*exp(-0.1296\*x)}  
  
g2 +  
 stat\_function(fun = func3, xlim=c(0,400), col = "blue", size = 1)

## Warning: Removed 4 rows containing missing values (geom\_point).

## Warning: Removed 264 rows containing missing values (geom\_path).



### Part 2 - Decomposition

This data was collected in Indonesia where oil palm leafs are put in between oil palm rows and left there to decompose.

I am first going to load my data from the .txt file I created and look at the first 5 observations too see what’s going on:

DATA=read.table("data/leaf-decomp-data.txt",h=T)  
head(DATA)

You may also want to try

View(DATA)

to see the whole data set.

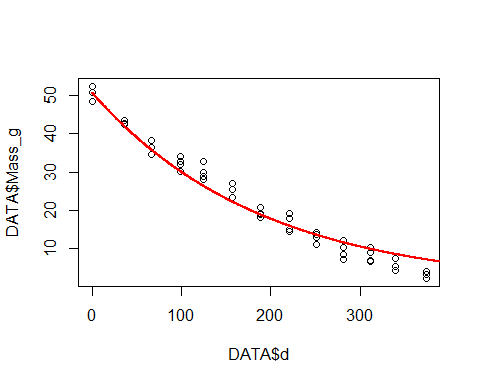
We now want to look at the first soil type (Clay Acrisol) at the first site, so we create a subset like this, using the pipe (%>%) from the tidyverse package. Then we check if this really worked.

DATA<-DATA%>%filter(Plot == "C1")  
unique(DATA$Plot)

## [1] C1  
## Levels: C1 C2 L1 L2

The next part is key! We are plotting our Data - days and mass, fit an exponential cuve and retrieve k, our decomposition rate:

xx <- seq(0,605, length=1500)  
plot(DATA$Mass\_g   
 ~ DATA$d)  
y <- DATA$Mass\_g  
x <- DATA$d  
fit <- nls(y ~ (50.81 \* 2.718^(-k \* x)), start=list(k=0.00001))  
lines(xx, predict(fit, data.frame(x=xx)), col="red", lwd=2, lty=1)



summary(fit)

##   
## Formula: y ~ (50.81 \* 2.718^(-k \* x))  
##   
## Parameters:  
## Estimate Std. Error t value Pr(>|t|)   
## k 0.0052707 0.0001223 43.1 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.582 on 51 degrees of freedom  
##   
## Number of iterations to convergence: 6   
## Achieved convergence tolerance: 5.804e-07

In the output above, we are given the k and some information on how well it fits the observed data.

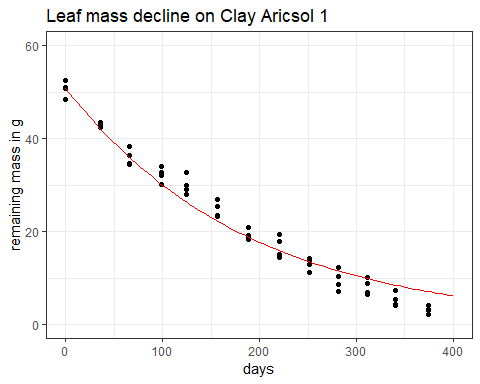
We made the plot using the basic plot package in R. However, in order to get a much nicer plot, I recommend using the ggplot 2 package. It’s already included in the tidyverse library we installed earlier.

We first define the function including k, which we get from above and define what data we want to use:

func1<- function(x) {50.81 \* 2.718^(-0.0052707 \* x)}  
dat1 = data.frame(x = DATA$d, y = DATA$Mass\_g)

Now we define the plot:

g1 <- ggplot(dat1,aes(x,y)) +  
 xlim(0,400) +  
 ylim(0,60) +  
 geom\_point(show.legend = FALSE) +  
 ggtitle("Leaf mass decline on Clay Aricsol 1") +  
 xlab("days") +  
 ylab("remaining mass in g") +  
 stat\_function(fun = func1, xlim=c(0,400), col = "red")  
g1

And this is what we get: 

To save your plot in a folder called figs within your project, we can use the cowplot package.

library(cowplot)  
ggsave("figs/oil-palm-decomp.png", plot = g1, width = 10, height = 10)