Laboratory 6: Site Index Curves in R

Felipe Bravo Oviedo

Universidad de Valladolid, Spain

In this Lab we will learn how to:

- Fit an anamorphic site index curve by using Hossfeld I function
- Graph the guide curve and add the lines of the site index curves

We will use also the knowledge we acquire in our previous labs and integrate our knowledge about site index curves development. As usual, first we must define our working directory. In this lab we do not need any data set.

```
# establishing the working directory

setwd('C:/your_desired_working_directoryR')
```

Now we should import the dataset and leaving only the data from the desired origine (in our case from stem analysis) by using this code:

```
# Importing data

DATA.SI<- read.csv2("DATA4_SiteIndex.csv", header=TRUE, sep=";", dec=".")

head(DATA.SI, 10)

# leaving only data from stem analysis (deleting H0_Age data)
        # H0_Age data come from plots were tree age and dominant height have
        # been measured while StemAnalysis data come from plots were some
        # trees have been cut in order to obtain discs where tree ring width and
        # age have been recorded
        # With the information from the discs the stem evolution has
        # been reconstructed

library(lattice)
DATA.SI <-subset(DATA.SI, type !="H0_Age")

# Explore data again to check if the H0_Age plot type has been removed
head(DATA.SI, 10)
```

Basic graph

In our previous labs we have knew how to draw different plot types. Now we will explore the distribution of dominant height with the age. The script to do that is the following:

```
# Plotting dominant age vs age

library(lattice)
xyplot(Height~Age,data=DATA.SI,groups=Plot,type="l", col="grey",
    main="Pinus sylvestris
    Alto Valle del Ebro, Spain", ylab="H0 (m)", xlab="Age (years)")
```

To obtain the following plot:

Pinus sylvestris
Alto Valle del Eb
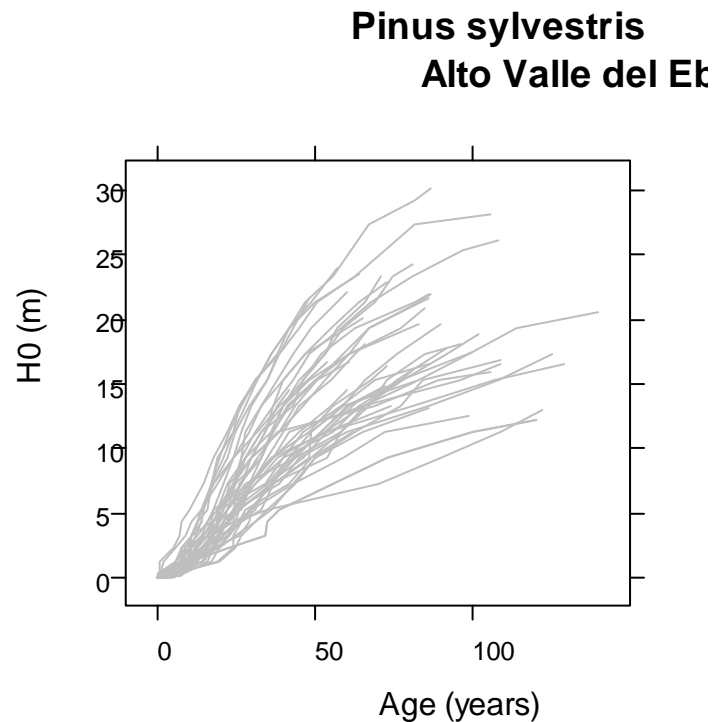
Exporting the graph

Now we can export the graph to our desired path and in our preferred format by using this code line:

```
# Exporting the graph
# png format and indicate size (width and height)

png('C:/your_desired_ouptut_directoryR', width = 683, height = 495)
```

Fitting anamorphic site index curves by using the guide curve method

We will use the Hossfeld I equation to illustrate this method. The Hossfeld I equation is the following:

$$H_0 = \frac{t^2}{(a + b*t)^2}$$

Where, H_0 is the dominant height and t is the stand age.

```
# SITE INDEX CURVES
# Fitting Hossfeld I model by guide curve method

DATA.SI <- subset(DATA.SI, Age !=0)
DATA.SI$vardep <- DATA.SI$Age/ DATA.SI$Height^0.5

# Explore data
head(DATA.SI, 10)

# guide 'curve' by linear regression
Hossfeld.lineal <- lm(DATA.SI$vardep ~ (DATA.SI$Age))
summary(Hossfeld.lineal)
```

To obtain the following output

```
Call:
lm(formula = DATA.SI$vardep ~ (DATA.SI$Age))

Residuals:
    Min      1Q  Median      3Q     Max
-5.2835 -1.9147 -0.1307  1.3788  9.8904

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.731933   0.179912   31.86   <2e-16 ***
DATA.SI$Age 0.176904   0.004046   43.73   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.62 on 582 degrees of freedom
Multiple R-squared:  0.7666,   Adjusted R-squared:  0.7662
F-statistic:  1912 on 1 and 582 DF,  p-value: < 2.2e-16
```

Graphing the guide curve

Now we will the graph the guide curve we fitted by using linear regression 8see the previous output). The script to obtain this graph is the following:

```
# Graphing the guide curve
X<-c(1:140)
Y<-c(X^2/(5.731933+0.176904*X)^2)
plot (X,Y,lwd=1, type="l", col = "red", lty=1, ljoin=10
    , main="Guide Curve  Hossfeld I"
    , ylab="Height (m)", xlab = "Age (year)")
```

To obtain the following graph:

Guide Curve
Hossfeld I
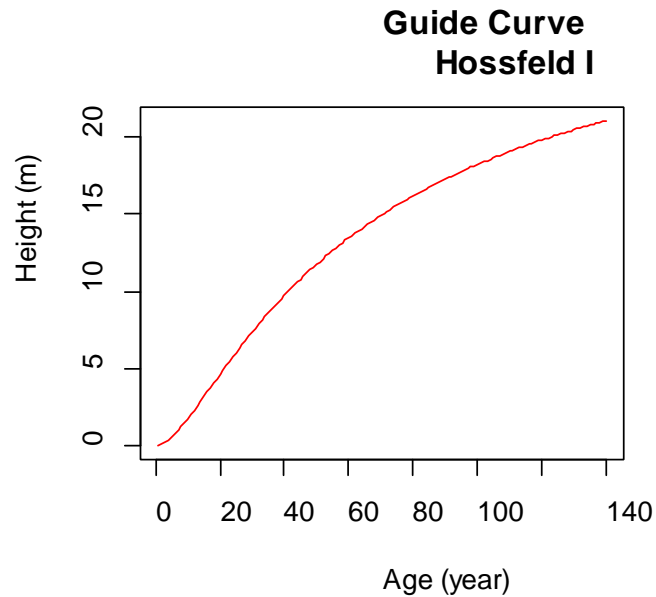
Graphing the Site Index Curves

For *Pinus sylvestris* L. in Spain, five different site index classes have been defined (H_0 equal 14, 17, 20, 23 and 26 m at the reference age, 100 years) With this value and fix as constant one of the Hossfeld I parameters we can obtain a set of site index curves. For instance, if we fix the slope constant (0.176904) we can obtain different independent terms (one per site index classes) by substituting the age for the reference age (100 years) and the dominant height by the site index classes centers at such age (H_0 equal 14, 17, 20, 23 and 26 m). Then we obtain the following independent terms (a) per each site index class:

- SI= 14 a = 9.03572419
- SI= 17 a = 6.5631625
- SI= 20 a = 4.67027977
- SI= 23 a = 3.16104141
- SI= 26 a = 1.92121351

Now we can proceed to graph the set site index curves with the following script

```
# SITE INDEX CURVES – Hosffeld I (with B₁ constant)

# define the length of the X-axis

X<-c(1:120)

# define the 5 site index curves

Y<-c(X^2/(9.03572419+0.176904*X)^2)
K<-c(X^2/(6.5631625+0.176904*X)^2)
Z<-c(X^2/(4.67027977+0.176904*X)^2)
G<-c(X^2/(3.16104141+0.176904*X)^2)
H<-c(X^2/(1.92121351+0.176904*X)^2)
```

```
# plotting the H site curve (the highest one)
plot (X,H,lwd=1, type="l", col = "red", lty=1, ljoin=10,
    main="Site index curves,
    Hossfeld I, b common",
            ylab="Height (m)", xlab = "Age (year)")
# add the other site index cuves
lines(X,K,lwd=1, col = "blue")
lines(X,Z,lwd=1, col = "green")
lines(X,G,lwd=1, col = "black")
lines(X,Y,lwd=1, col = "yellow")

        # lty (Line types) can either be specified as an integer
        # (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash,
        # 6=twodash)
        # or as one of the character strings "blank", "solid", "dashed", "dotted",
        # "dotdash", "longdash", or "twodash",
        # where "blank" uses □invisible lines□ (i.e., does not draw them).
        # lty also works with lines and segments
```

To obtain the following graph:

**Site index curves,
Hossfeld I, b con**
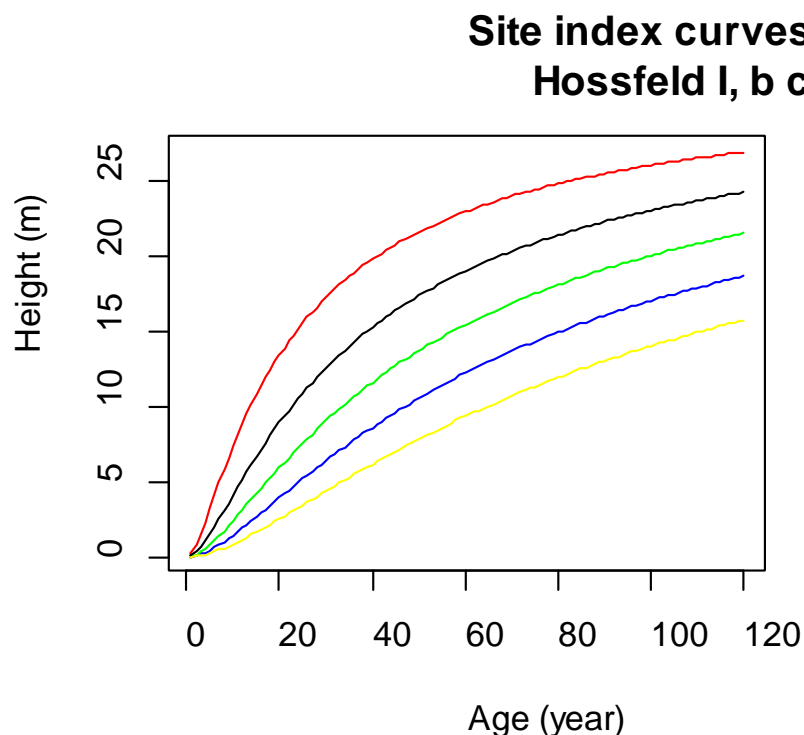
An alternative is to fix the independent term (5.731933) and modify the slope to force the lines to pass through the defined site index values.

- SI= 14 b = 0.20994191
- SI= 17 b = 0.1852163
- SI= 20 b = 0.16628747
- SI= 23 b = 0.15119508
- SI= 26 b = 0.13879680

Now the graph can be obtained with the following script

```
# SITE INDEX CURVES – Hosffeld I (with B0 constant)

# define the length of the X-axis
X<-c(1:120)

# define the 5 site index curves
Q<-c(X^2/(5.731933+0.20994191 *X)^2)
W<-c(X^2/(5.731933+0.1852163 *X)^2)
E<-c(X^2/(5.731933+0.16628747 *X)^2)
R<-c(X^2/(5.731933+0.15119508 *X)^2)
U<-c(X^2/(5.731933+0.13879680 *X)^2)

# plotting the H site curve (the highest one)
plot (X, U, lwd=1, type="l", col = "red", lty=1, ljoin=10,
    main="Site index curves,
    Hossfeld I, a common"
    , ylab="Height (m)", xlab = "Age (year)")

# add the other site index cuves
lines(X,W,lwd=1, col = "blue")
lines(X,E,lwd=1, col = "green")
lines(X,R,lwd=1, col = "black")
lines(X,Q,lwd=1, col = "yellow")
```
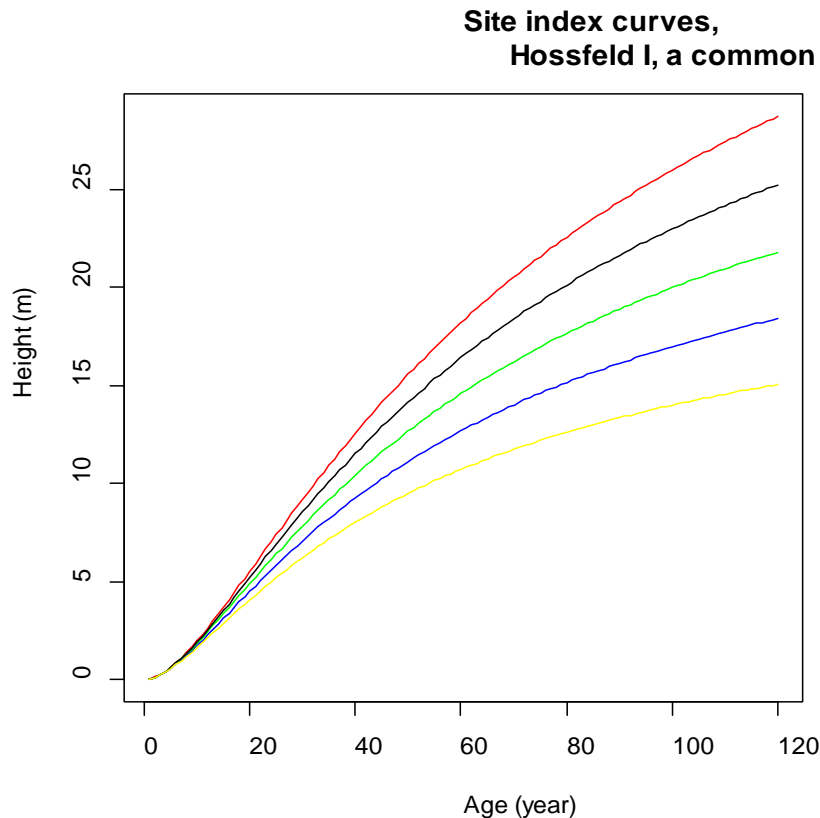
To obtain the following graph:

Site index curves,
Hossfeld I, a common

Superimposing the stem analysis to the Site Index Curves

In case we would like to superimpose the stem analysis to a site index curves (in this case the Hossfeld I with slope, B_1, constant) we can obtain it with the following script:

```
# Superimposing the stem analysis to the Site Index Curves
xyplot(Height~Age,data=DATA.SI,groups=Plot,type="l", col="grey",
    main="Stem analysis (Ps), Alto Valle del Ebro (Spain)
    Hossfeld I b common", ylab="Height (m)", xlab="Age (year)",
     panel=function(x, y, ...){
     panel.xyplot(x, y, ...)
     panel.lines(X, K, col=1, lwd=2)
     panel.lines(X, Z, col=1, lwd=2)
     panel.lines(X, G, col=1, lwd=2)
     panel.lines(X, Y, col=1, lwd=2)
     panel.lines(X, H, col=1, lwd=2)})
```

Then we will obtain the following graph:

Stem analysis (Ps), Alto Valle del Ebro (Spain)
Hossfeld I b common
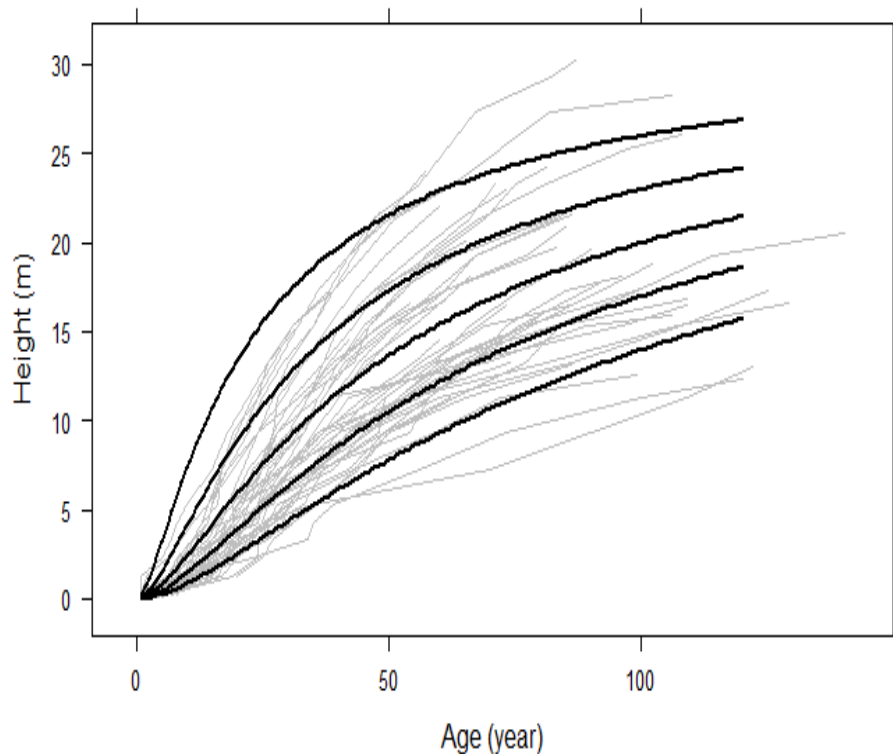
Full script

You can copy and paste into your R environment (changing your working directory)

```
# establishing the working directory

setwd('C:/your_desired_working_directoryR')

# Importing data

DATA.SI<- read.csv2("DATA4_SiteIndex.csv", header=TRUE, sep=";", dec=".")

head(DATA.SI, 10)

# leaving only data from stem analysis (deleting H0_Age data)
        # H0_Age data come from plots were tree age and dominant height have
        # been measured while Stem Analysis data come from plots were some
        # trees have been cut in order to obtain discs where tree ring width and
        # age have been recorded
        # With the information from the discs the stem evolution has
        # been reconstructed

library(lattice)
DATA.SI <-subset(DATA.SI, type !="H0_Age")
```

```
# Explore data again to check if the H0_Age plot type has been removed
head(DATA.SI, 10)

# Plotting dominant age vs age

library(lattice)
xyplot(Height~Age,data=DATA.SI,groups=Plot,type="l", col="grey",
    main="Pinus sylvestris
    Alto Valle del Ebro, Spain", ylab="H0 (m)", xlab="Age (years)",)

# Exporting the graph
# png format and indicate size (width and height)

png('C:/your_desired_ouptut_directoryR', width = 683, height = 495)

# SITE INDEX CURVES
# Fitting Hossfeld I model by guide curve method

DATA.SI <- subset(DATA.SI, Age !=0)
DATA.SI$vardep <- DATA.SI$Age/ DATA.SI$Height^0.5

# Explore data
head(DATA.SI, 10)

# guide 'curve' by linear regression
Hossfeld.lineal <- lm(DATA.SI$vardep ~ (DATA.SI$Age))
summary(Hossfeld.lineal)

# Graphing the guide curve
X<-c(1:140)
Y<-c(X^2/(5.731933+0.176904*X)^2)
plot (X,Y,lwd=1, type="l", col = "red", lty=1, ljoin=10,
    main="Guide Curve
    Hossfeld I",
              ylab="Height (m)", xlab = "Age (year)")

# SITE INDEX CURVES – Hosffeld I (with B1 constant)

# define the length of the X-axis

X<-c(1:120)

# define the 5 site index curves

Y<-c(X^2/(9.03572419+0.176904*X)^2)
K<-c(X^2/(6.5631625+0.176904*X)^2)
Z<-c(X^2/(4.67027977+0.176904*X)^2)
G<-c(X^2/(3.16104141+0.176904*X)^2)
H<-c(X^2/(1.92121351+0.176904*X)^2)

# plotting the H site curve (the highest one)
```

```r
plot (X,H,lwd=1, type="l", col = "red", lty=1, ljoin=10,
    main="Site index curves,
    Hossfeld I, b common",
                ylab="Height (m)", xlab = "Age (year)")
# add the other site index cuves
lines(X,K,lwd=1, col = "blue")
lines(X,Z,lwd=1, col = "green")
lines(X,G,lwd=1, col = "black")
lines(X,Y,lwd=1, col = "yellow")


        # lty (Line types) can either be specified as an integer
        # (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash,
        # 6=twodash)
        # or as one of the character strings "blank", "solid", "dashed", "dotted",
        # "dotdash", "longdash", or "twodash",
        # where "blank" uses □invisible lines□ (i.e., does not draw them).
        # lty also works with lines and segments

# SITE INDEX CURVES – Hosffeld I (with B₀ constant)

# define the length of the X-axis
X<-c(1:120)

# define the 5 site index curves
Q<-c(X^2/(5.731933+0.20994191 *X)^2)
W<-c(X^2/(5.731933+0.1852163 *X)^2)
E<-c(X^2/(5.731933+0.16628747 *X)^2)
R<-c(X^2/(5.731933+0.15119508 *X)^2)
U<-c(X^2/(5.731933+0.13879680 *X)^2)

# plotting the H site curve (the highest one)
plot (X,U,lwd=1, type="l", col = "red", lty=1, ljoin=10,
    main="Site index curves,
    Hossfeld I, b common",
                ylab="Height (m)", xlab = "Age (year)")
# add the other site index cuves
lines(X,W,lwd=1, col = "blue")
lines(X,E,lwd=1, col = "green")
lines(X,R,lwd=1, col = "black")
lines(X,Q,lwd=1, col = "yellow")

# Superimposing the stem analysis to the Site Index Curves
xyplot(Height~Age,data=DATA.SI,groups=Plot,type="l", col="grey",
     main="Stem analysis (Ps), Alto Valle del Ebro (Spain)
     Hossfeld I b common", ylab="Height (m)", xlab="Age (year)",
       panel=function(x, y, ...){
       panel.xyplot(x, y, ...)
       panel.lines(X, K, col=1, lwd=2)
       panel.lines(X, Z, col=1, lwd=2)
       panel.lines(X, G, col=1, lwd=2)
       panel.lines(X, Y, col=1, lwd=2)
       panel.lines(X, H, col=1, lwd=2)})
```

Let's try now the nonlinear regression approach

For *Pinus sylvestris* L. in Spain, five different site index classes have been defined (H_0 equal 14, 17, 20, 23 and 26 m at the reference age, 100 years) With this value and fix as constant

```
# Defining the Hossfeld model as an object to fit by nonlinear regression
Hossfeld <- Height~Age^2/(a+b*Age)^2
# Fit the model by non-linear regression.
# Start or seed values are the parameters from the linear regression
nlmod <- nls(Hossfeld, data=DATA.SI, start=list(a=5.7319, b=0.17166))
# Obtain the results from the regression
summary(nlmod)
```

Check the results and repeat the graphical analysis we performed with the parameters obtained by nonlinear regression.