Basic Graphs RStudio and R Markdown

Course Instructor: Dr. Suborna Ahmed

# Basic Graphs in R/RStudio/R Markdown

In this module you will learn how to create basic graphs in R. Scatter plots are used to show the association between two numerical variables. In the horizontal axis (the x-axis) we place the values that correspond to the explanatory variable, along the vertical axis (the y-axis) we place the values that correspond to the response variable. The pattern of the cloud of points that we observe in scatters plots gives us an idea about the association between the two variables and if it is positive, negative, or absent.

# Add a coded chunk and check your Scatter plots

Add a codded chunk and check your directory getwd()where you kept this “Basic Graphs RStudio and R Markdown.Rmd” file. Keep all data files and this .RMD file in one folder.

# Creating plots

We can create a plot in R by using the function plot() and specifying each of the variables that we want to compare inside the parenthesis. This will create a graph with some basic information about the two variables that we are comparing. To add more details to our plots and make them more visually appealing there are multiple functions that we can add to our code.

# Label x-axis and y-axix: xlab and ylab

We can change the text of the x-axis label or the y-axis label by using the functions xlab() or ylab(). The labels that we want to add should be included in quotes inside the parentheses of each function. In these labels we can also specify the units that were used to measure each of our variables (e.g years, meters, etc).

# Type of graph: type

We use the type to specify the type of graph that we want to create. There are several values possible for this option including “p” for points, “l” for lines, “b” for both points and lines, or “c” for empty points joined by lines.

# Symbol and colour of data points:

In scatter plots each observation is represented as a point. We can change the character or symbol of our data points by using the pch function. To plot a specific character, we have to specify the number of the symbol that we want to use in this function (e. g. pch= 16). In Fig.1 you will find some of the characters that are available for this function. To change the colour of the data points, use the col function. You can change the colour of the data points by writing the name of the colour in quotes using this function (e. g. col=“black”). To find a complete list of the colours available in R go to [this site.](http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf)

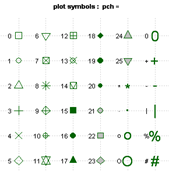


Figure 1. Plot symbols for data points using the pch function.

# Axis limit: xlim and ylim

We can adjust the limits of the x-axis and y-axis by using xlim and ylim functions. By adjusting these limits according to your data, you will be able to see all the data points inside of the graph area. The first number corresponds to the lower limit and the second to the upper limit.

# Title: main

Use this function to add a title to the graph. The name should be inside quotes (e. g. main= “Height trend of leading species”).

# Size control: cex

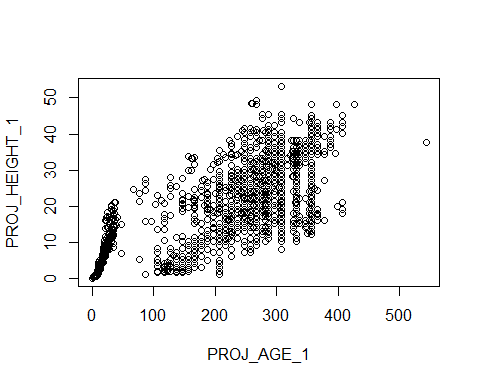
This function allows us to change the size of the data points, labels, title, and axes. Use cex to change the size of data points, cex.lab to change the size of titles in axes, cex.axis to change the size of numbers in the axes, and cex.main to change the size of the title of the graph.

# Simple graph

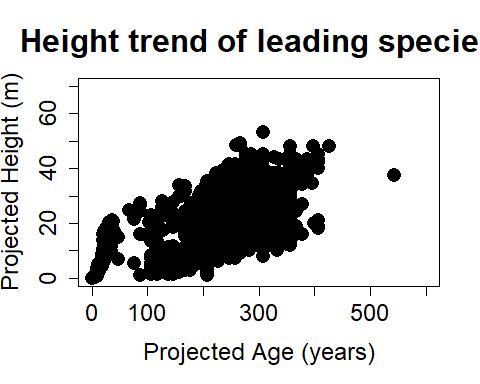
# Check Current directory  
getwd()

## [1] "C:/Syd/UBC/Teaching/GAA FRST 232"

# Importing VRI data  
vri\_data <- read.csv("VRI\_data.csv", header = TRUE)  
  
# Attach the data  
attach(vri\_data)  
  
# Creating graph without controlling any element  
plot(PROJ\_AGE\_1,PROJ\_HEIGHT\_1)



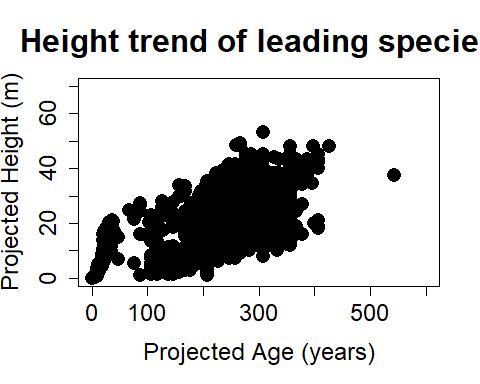
# Insert important elements in the plot fucntion  
plot(vri\_data[,3], vri\_data[,4],   
 xlab="Projected Age (years)",   
 ylab="Projected Height (m)",   
 type = "p",   
 pch = 16,   
 col = "black",   
 ylim=c(0,70),   
 xlim=c(0,600),   
 cex=2,   
 cex.lab=1.5,   
 cex.axis=1.5,   
 main="Height trend of leading species",   
 cex.main=2)



# Define parameters and call inside the plot function

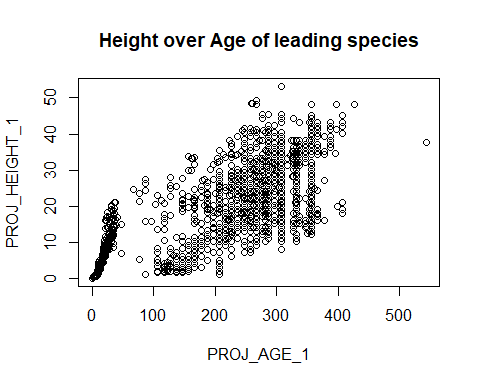
Sometimes it is more convenient to define the parameters that we what to add to our graph first, and then call them into the plot function. This is useful when we want to make multiples graphs with different information, and we do not want to repeat multiple times parts of our code. To create a parameter, use name\_of\_parameter= parameter\_description. Once you have created the parameter you can use it with other functions. For example, we can define the labels of our axes as xlabel= “Projected Height (m)” and ylabel= “Projected Age (years)” and then call them into the plot function as plot(PROJ\_AGE\_1,PROJ\_HEIGHT\_1,xlab=xlabel, ylab=ylabel).

# Define All values  
  
ylabel="Projected Height (m)"  
xlabel="Projected Age (years)"  
ylm=c(0,70)  
xlm=c(0,600)  
ldcex=2 # size of points  
cxlb=1.5 # axis level - fontsize  
cxaxis=1.5 # axis fontsize  
maincx=2 # title font  
  
titel="Height trend of leading species"  
  
  
# Call the above paramters inside the plot fucntion  
plot(PROJ\_AGE\_1,PROJ\_HEIGHT\_1,   
 xlab=xlabel,   
 ylab=ylabel,   
 type = "p",   
 pch = 16,   
 col = "black",   
 ylim=ylm,   
 xlim=xlm,   
 cex=ldcex,   
 cex.lab=cxlb,   
 cex.axis=cxaxis,   
 main=titel,   
 cex.main=maincx)

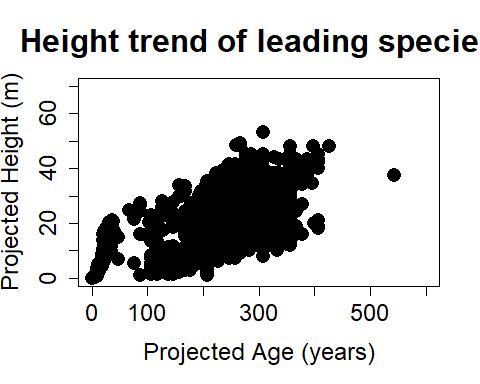


# Plotting a graph

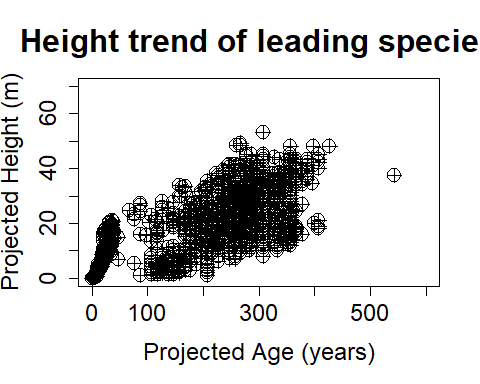
# plain graph  
plot(PROJ\_AGE\_1,PROJ\_HEIGHT\_1, main="Height over Age of leading species")



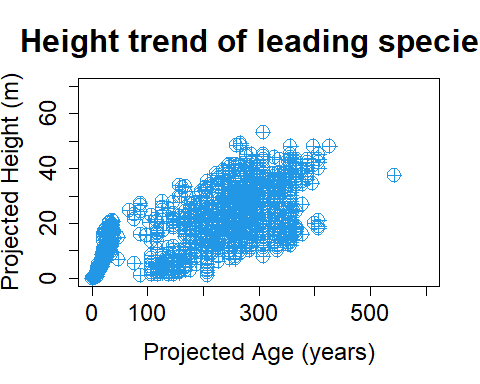
# Add detail  
age=PROJ\_AGE\_1  
ht=PROJ\_HEIGHT\_1  
ylabel="Projected Height (m)"  
xlabel="Projected Age (years)"  
ylm=c(0,70)  
xlm=c(0,600)  
  
cxlb=1.5 # axis level - fontsize  
cxaxis=1.5 # axis fontsize  
maincx=2 # title font  
  
  
ldcex=2 # size of points  
ld=4  
  
# 1st graph  
plot(age, ht, xlab=xlabel, ylab=ylabel, type = "p", pch = 16, col = "black", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, main="Height trend of leading species", cex.main=maincx)



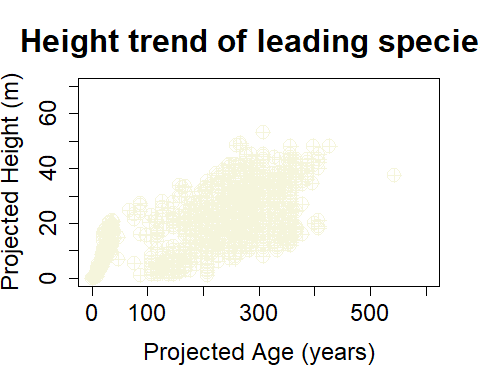
# change symbol on the previous graph  
plot(age, ht, xlab=xlabel, ylab=ylabel, type = "p", pch = 10, col = "black", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, main="Height trend of leading species", cex.main=maincx)



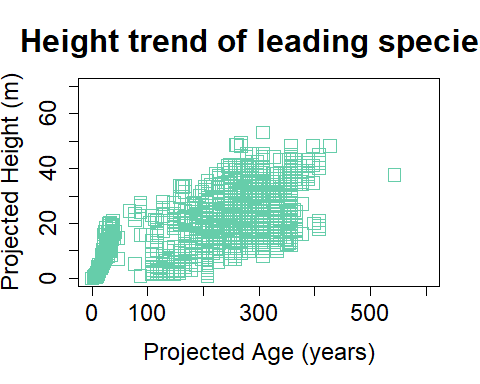
# change colour on the previous graph  
  
plot(age, ht, xlab=xlabel, ylab=ylabel, type = "p", pch = 10, col = 84, ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, main="Height trend of leading species", cex.main=maincx)



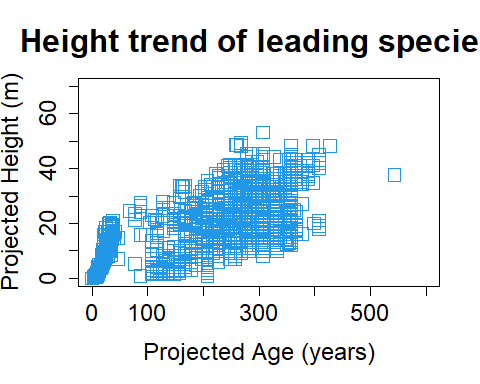
plot(age, ht, xlab=xlabel, ylab=ylabel, type = "p", pch = 10, col = "beige", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, main="Height trend of leading species", cex.main=maincx)



plot(age, ht, xlab=xlabel, ylab=ylabel, type = "p", pch = 22, col='#66CDAA', ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, main="Height trend of leading species", cex.main=maincx)



plot(age, ht, xlab=xlabel, ylab=ylabel, type = "p", pch = 22, col=68, ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, main="Height trend of leading species", cex.main=maincx)

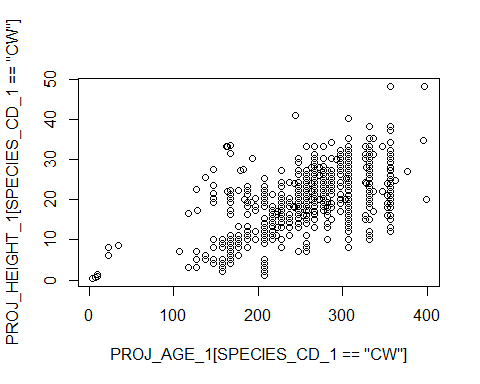


# Plotting a subset of data and plotting two dataset on the same graph

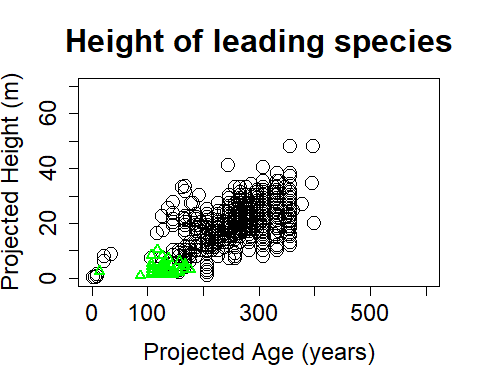
To plot a subset of data we should specify the data frame that we want to use as well as the variable from where we want to take the subset of data. To do this we should define x and y axes by using plot(name\_of\_dataframe[name\_of\_column == “name\_of\_subset”]), this will generate a graph with default attributes. To add specific features to our plot we can use any of the functions previously described inside of the parenthesis separated by commas plot(name\_of\_dataframe[name\_of\_column1 == “name\_of\_subset”], name\_of\_dataframe[name\_of\_column2 == “name\_of\_subset”] type = “p”, pch = 1, col = “black”, main=“Height of leading species”).

We can plot two datasets on the same graph by using the points() function. First, we have to plot a subset of the data such as it was described above. Then we add points() and define the x and y axes for this dataset by using plot(name\_of\_dataframe[name\_of\_column1 == “name\_of\_subset2”], name\_of\_dataframe[name\_of\_column2 == “name\_of\_subset2”]. Alternatively, you can also define each subset of data first and then use these vectors to create the plot. In both cases, make sure that you run both chunks of code at the same time to plot the graph.

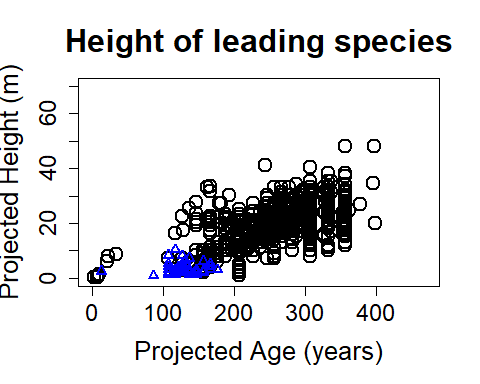
# plot a subset: subset the data and plot together  
plot(PROJ\_AGE\_1[SPECIES\_CD\_1 =='CW'],PROJ\_HEIGHT\_1[SPECIES\_CD\_1 =='CW'])



# plot a subset and two dataset on the same graph  
  
plot(PROJ\_AGE\_1[SPECIES\_CD\_1 =='CW'],PROJ\_HEIGHT\_1[SPECIES\_CD\_1 =='CW'], type = "p", pch = 1, col = "black", main="Height of leading species", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, cex.main=maincx, xlab=xlabel, ylab=ylabel)  
points(PROJ\_AGE\_1[SPECIES\_CD\_1 =='PLC'], PROJ\_HEIGHT\_1[SPECIES\_CD\_1=='PLC'], type = "p", pch = 2, col = "green", lwd=2)

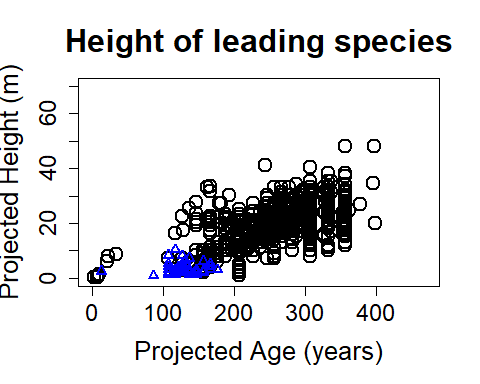


# plot subset: alternatevely  
  
# separating data by species  
d1<- subset(vri\_data, (SPECIES\_CD\_1 =='CW'))  
d2<- subset(vri\_data, (SPECIES\_CD\_1 =='PLC'))  
  
# define elements  
  
age=3 # age column number  
ht=4 # heightt column number  
ylabel="Projected Height (m)"  
xlabel="Projected Age (years)"  
ylm=c(0,70)  
xlm=c(0,470)  
  
cxlb=1.6 # axis level - fontsize  
cxaxis=1.5 # axis fontsize  
maincx=2 # title font  
  
ldcex=2 # size ofpoints  
ld=2  
  
plot(d1[,age],d1[,ht], type = "p", pch = 1, col = "black", main="Height of leading species", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, cex.main=maincx, xlab=xlabel, ylab=ylabel, lwd=ld)  
points(d2[,age],d2[,ht], type = "p", pch = 2, col = "blue", lwd=ld)



# Alternatively: Create the subset and then plot

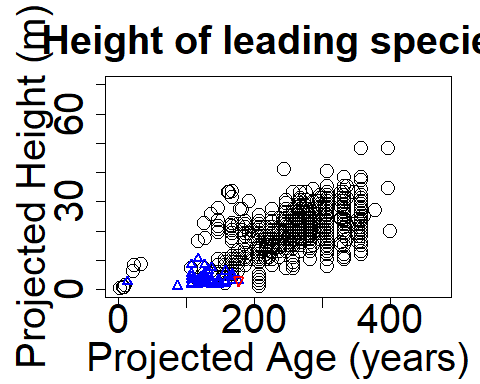
# plot subset: alternatevely  
  
# separating data by species  
d1<- subset(vri\_data, (SPECIES\_CD\_1 =='CW'))  
d2<- subset(vri\_data, (SPECIES\_CD\_1 =='PLC'))  
  
# define elements  
age=3 # age column number  
ht=4 # heightt column number  
ylabel="Projected Height (m)"  
xlabel="Projected Age (years)"  
ylm=c(0,70)  
xlm=c(0,470)  
  
cxlb=1.6 # axis level - fontsize  
cxaxis=1.5 # axis fontsize  
maincx=2 # title font  
  
ldcex=2 # size ofpoints  
ld=2  
  
plot(d1[,3],d1[,4], type = "p", pch = 1, col = "black", main="Height of leading species", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, cex.main=maincx, xlab=xlabel, ylab=ylabel, lwd=ld)  
points(d2[,3],d2[,4], type = "p", pch = 2, col = "blue", lwd=ld)



# Controlling margins

We can plot multiple graphs in a single plot by using the par() function. This function allows us to set multiple graphical parameters by using different arguments. We can specify, for instance, the number of subplots we need by using mfrow() where we specify first the number of rows and then the number of columns we want (mfrow (#rows, #columns)). To define the margins of our plot we use the main() function. In this function, we must give the four values that we want as margin space in the bottom, left, top, and right parts of the chart, respectively. Margin space is given in inches.

# Defining parameters  
ylabel="Projected Height (m)"  
xlabel="Projected Age (years)"  
ylm=c(0,70)  
xlm=c(0,470)  
  
cxlb=2.5 # axis level - fontsize  
cxaxis=2.5 # axis fontsize  
maincx=2.5 # title font  
  
age=3 # age column number  
ht=4 # heightt column number  
ldcex=2 # size ofpoints  
ld=2  
  
# Finding the height value at the maximum age   
maxd1<- max(d2[,age]) # maximum age  
m1<-subset(d2, PROJ\_AGE\_1==maxd1) # height value at the maximum age  
  
  
# Setting margin  
par(mfrow=c(1,1),mai=c(0.9,1.1,0.8,0.3), cex=1.0 ) # mai is the margin  
  
# Creating a single graph  
plot(d1[,age],d1[,ht], type = "p", pch = 1, col = "black", main="Height of leading species", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, cex.main=maincx, xlab=xlabel, ylab=ylabel)  
points(d2[,age],d2[,ht], type = "p", pch = 2, col = "blue", lwd=ld)  
points(m1[,age],m1[,ht], type = "p", pch = 6, col = "red", lwd=ld)



# we can show the mean value as well  
mean(d2[,age])

## [1] 121.8132

# exercise: show the mean value in agraph