Barplot in R Markdown

Dr. Suborna Ahmed

# Preparing directory and data setup

In this lecture, we will be creating side-by-side graphs for two different species using the vri.csv file. For doing this use the subset() and par() functions such as were explained in previous lectures.

# VRI data  
  
vri\_data<-read.csv("VRI\_data.csv",header = T)  
  
attach(vri\_data)  
  
# separating data by species  
d1<- subset(vri\_data, (SPECIES\_CD\_1 =='CW'))  
d2<- subset(vri\_data, (SPECIES\_CD\_1 =='PLC'))

# Add one legend for multiple graphs

We can add a legend for multiple graphs by using the legend() function. In this case, we have to use the last graph created as a reference to add the coordinates of the legend. Make sure that you give appropriate margin space in the graph for adding the legend using the mai= option in the par() function. Use xpd=NA to specify that we want the legend outside of the graph area.

ylabel="Projected Height (m)"  
xlabel="Projected Age (years)"  
ylm=c(0,60)  
xlm=c(0,470)  
  
cxlb=1.1 # axis level - fontsize  
cxaxis=1.1 # axis fontsize  
maincx=2.5 # title font  
  
row1=3 # age column number  
row2=4 # ht column number  
ldcex=1 # size ofpoints  
ld=1  
maxd1<- max(d2[,row2])  
m1<-subset(d2, PROJ\_HEIGHT\_1==maxd1)  
maxd2<- max(d1[,row2])  
m2<-subset(d1, PROJ\_HEIGHT\_1==maxd2)  
#names(vri\_data)  
  
# creating a permanent image file  
png("One\_legend.png", width=5, height=5, units = 'in', res=1200)  
par(mfrow=c(1,2),mai=c(3,0.9,0.2,0.3), cex=1.0, xpd=NA) # mai is the margin  
  
# graph 1  
plot(d1[,row1],d1[,row2], type = "p", pch = 20, col = "blue", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, xlab=xlabel, ylab=ylabel)  
points(m2[,row1],m2[,row2], type = "p", pch =11, col = "red", lwd=ld)  
  
# graph 2  
plot(d2[,row1],d2[,row2], type = "p", pch = 20, col = "black", ylim=ylm, xlim=xlm, cex=ldcex, cex.lab=cxlb, cex.axis=cxaxis, xlab=xlabel, ylab=ylabel)  
points(m1[,row1],m1[,row2], type = "p", pch =11, col = "red", lwd=ld)  
  
legend(-700, -35, ncol=1 ,c("Projected height: Species is 'CW'", "Projected height: Species is 'PLC'", "Maximum Projected Height"), col = c("blue", "black", "red"), pch = c(20, 20, 11), cex=1, pt.cex=ld)  
  
graphics.off()

# Creating barplots

For this section we will create a bar plot using the trees.txt file. Use the read.table() function for importing this file into R. Then:

1. Determine the type of variables that you have (categorical or numerical). In this case we want to know what type of variable is “CC”. Use the class() function for checking on this.
2. Use table() to create a vector with the counts of the number of items that you have in the CC column.
3. Use barplot() function to create a bar plot. Use the vector that you create with the counts as a file and modify graph limits, colours, and labels if desired.

# .txt data file  
trees <- read.table("trees.txt", header = TRUE)  
# checking the data:  
dim(trees)

## [1] 50 7

names(trees)

## [1] "ID" "Date" "Species" "CC" "NNT" "DBH" "Height"

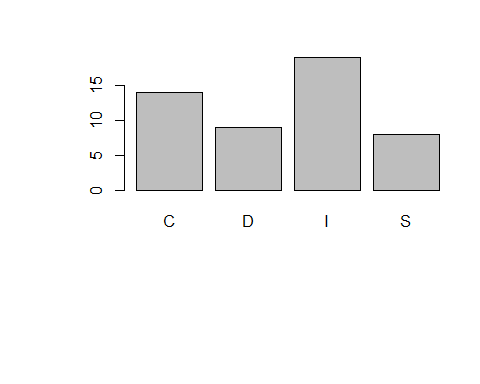
# attached all variables  
attach(trees)  
  
# crown class variable and create a table  
class(CC)

## [1] "character"

counts <- table(CC)  
counts

## CC  
## C D I S   
## 14 9 19 8

#barplot 1  
par(mfrow=c(1,1),mai=c(1,0.9,0.2,0.3), cex=1.0, oma=c(5,2,2,0), xpd=NA) # mai is the margin  
barplot(counts)



graphics.off()  
  
#barplot 2  
graphics.off()  
par(mfrow=c(1,1),mai=c(1,0.9,0.2,0.3), cex=1.0, oma=c(5,2,2,0), xpd=NA) # mai is the margin  
barplot(counts, ylim = c(0, 20))  
  
#barplot 3  
graphics.off()  
par(mfrow=c(1,1),mai=c(1,0.9,0.2,0.3), cex=1.0, oma=c(5,2,2,0), xpd=NA) # mai is the margin  
barplot(counts, ylim = c(0, 20), ylab = "Frequency", xlab = "Crown Classes")  
  
detach(trees)

# VRI data bar chart

We can also create bar plots by selecting a categorical variable and using the plot() function. In this case, R will automatically create a bar plot with the selected categorical variable (e.g. plot (SPECIES\_CD\_1) ). To change the direction of the numbers in the y axis use las=1 into barplot(). To change the names in the x axis use names.arg= c(“name\_variable1”, “name\_variable2”). This will change the names that you have for each column in the x axis. To modify the font size use cex.names=.

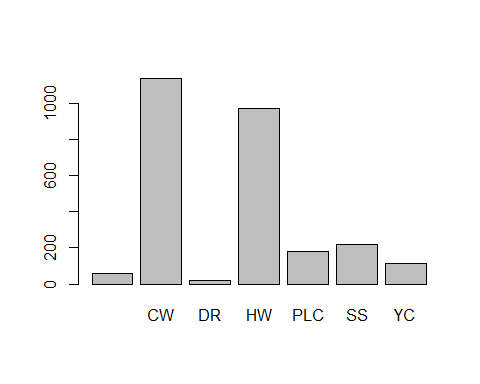
attach(vri\_data)

## The following objects are masked from vri\_data (pos = 3):  
##   
## LIVE\_VOL\_PER\_HA\_SPP1\_125, OBJECTID, PROJ\_AGE\_1, PROJ\_HEIGHT\_1,  
## SPECIES\_CD\_1

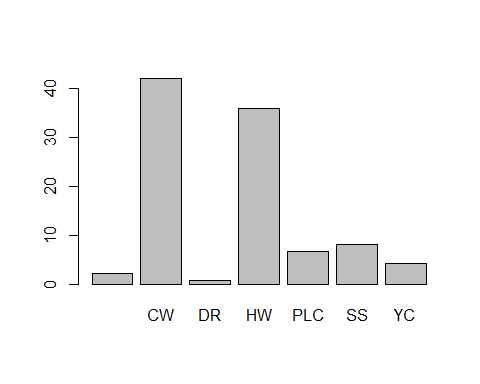
class(SPECIES\_CD\_1)

## [1] "character"

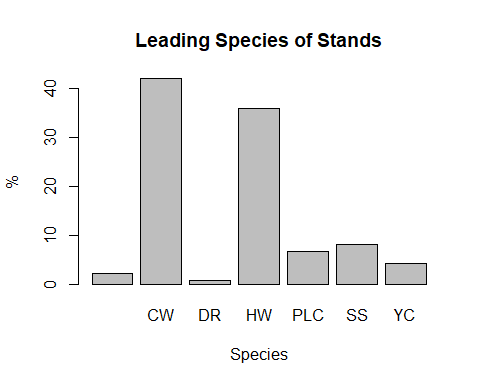
#barplot 1  
#coverting the character variable as a factor to plot it  
  
plot(factor(SPECIES\_CD\_1))



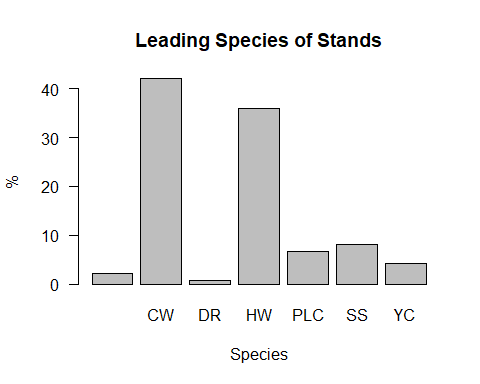
#barplot 2  
leading\_spec <- table(SPECIES\_CD\_1)  
percent\_spec <- (table(SPECIES\_CD\_1)\*100)/2700  
barplot(percent\_spec)



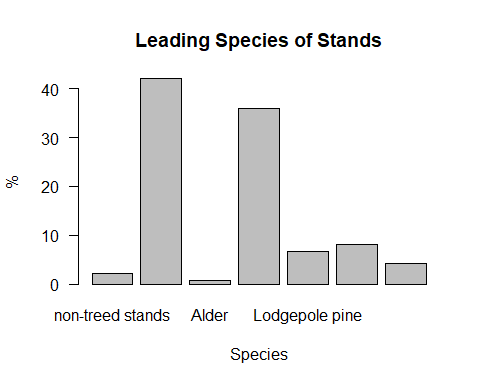
#barplot 3  
barplot(percent\_spec, main="Leading Species of Stands", xlab="Species", ylab="%")



#barplot 4: rotate y axis values:  
  
barplot(percent\_spec, main="Leading Species of Stands", xlab="Species", ylab="%", las=1)



#barplot 5  
# set the names of categories under x axis:  
barplot(percent\_spec, main="Leading Species of Stands", xlab="Species", ylab="%", las=1, names.arg=c("non-treed stands", "Western Red Cedar", "Alder", "Western Hemlock", "Lodgepole pine", "Sitka Spruce", "Yellow Cedar"))



# barplot 6: reduce the names to adjust the space  
barplot(percent\_spec, main="Leading Species of Stands", xlab="Species", ylab="%", las=1, names.arg=c("Non-treed", "Western R. Cedar", "Alder", "Western Hemlock", "L. pine", "Sitka Spruce", "Yellow Cedar"), ylim=c(0,60), cex.axis=1.5, cex.lab=1.5, cex.names=0.7)

