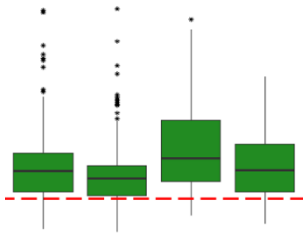
# An introduction to R coding




## Asad Prodhan

BScAg (Hons) MS *BAU* MSc *Syd.* PhD *W.Aust.* JSPS Fellow

R is a popular programming language in data science. Dealing with any sort of data inevitably involves the following three tasks: i) pre-analysis processing of the raw data, ii) careful selection of suitable statistical tests, and iii) finally visualizing the data in a way that captures the take-home message. In this tutorial, we aim to gain some basic skills to perform these three tasks using R. For each task, we will look at a real-world example and go through how to write R codes to accomplish them. We start with the data-visualisation task first.

**What does data visualisation do?**

Data visualisation is a graphical representation of the actual data. It makes it easier to see the underlying patterns present in the data. For example, box plot, bar graph, line graph etc.

R has various commands that allow for making various types of plots or graphs. The special advantage of using R is that it also comes with many other commands that can be used to further polish the graphs or write extra information on the graphs. As a result, you get full control on how you want to present your results and what you want to highlight!

However, these R commands are bundled together, which is known as 'package' or 'library'. You need to know which R package has the commands to make the graph of your choice. For example, in this tutorial, we want to make a box plot using a real experimental data-set. First, we google to find out which R package can make a boxplot. Google suggests 'ggplot2'.

OK. Let's use 'ggplot2' and make a boxplot out of our data-set'.

Now, how do we do this? Well, to do so, we need to make the "ggplot2" package and our data-set available in R Studio.

So, let's install and upload the 'ggplot2' package in R Studio. You need to install it only once on your computer. The following command will check your installed R packages. If ggplot2 is not installed yet, it will install and upload it; If already installed, then it will avoid re-installation and just upload it.

The 'tinytex' package is for producing a pdf file of this tutorial. So, you can ignore it for now.

### Installing R packages

```
if (!require(ggplot2)) {
    install.packages("ggplot2")
    require(ggplot2)
}
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
if (!require(tinytex)) {
    install.packages("tinytex")
    require(tinytex)
}
```

```
## Loading required package: tinytex
```

### Uploading R packages

Upload 'ggplot2'. You need to upload it every time you will be using it.

```
library(ggplot2)
library(tinytex)
```

Now, we have the "ggplot2" available in R Studio. Next, we need to get our data-set available in R Studio.

### Uploading (also called 'reading') the data sheet in R Studio:

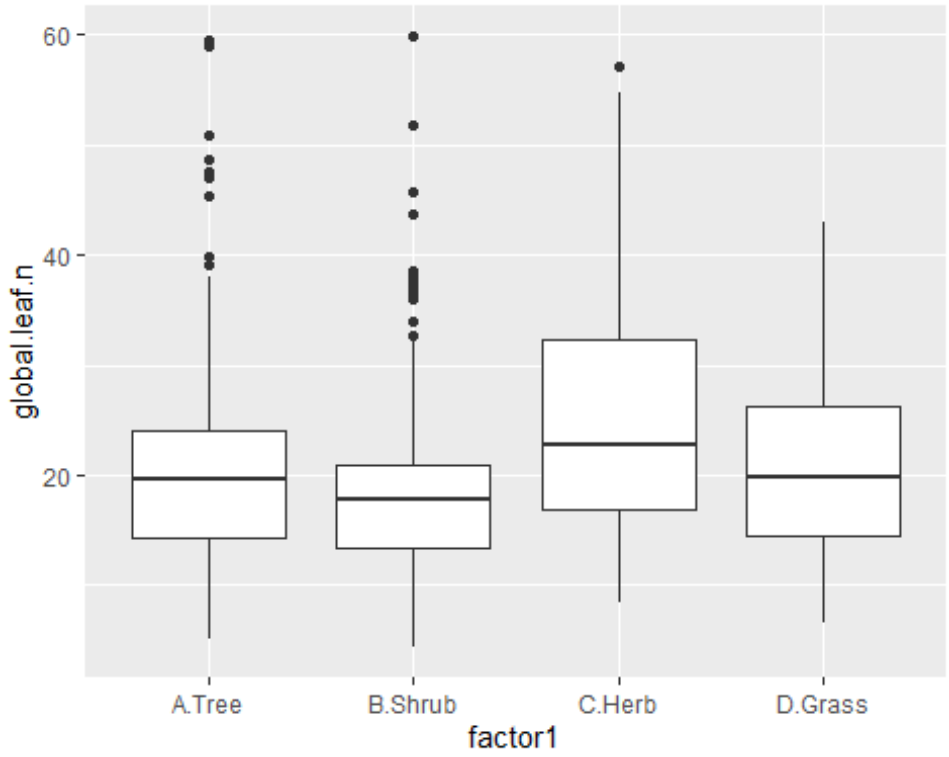
```
df <- read.csv('npdat6.csv') # we name our data-set as 'df'
head(df)
```

```
##   factor1                  species global.leaf.n
## 1  A.Tree         Erythrina caffra          59.5
## 2  A.Tree       Cladostemon kirkii          59.0
## 3  A.Tree    Erythrina poeppigiana          50.8
## 4  A.Tree Erythrophleum lasianthum          48.6
## 5  A.Tree         Dalbergia armata          47.6
## 6  A.Tree      Wrightia natalensis          47.0
```

At this point, both "ggplot2" and the data-set are available in R Studio.

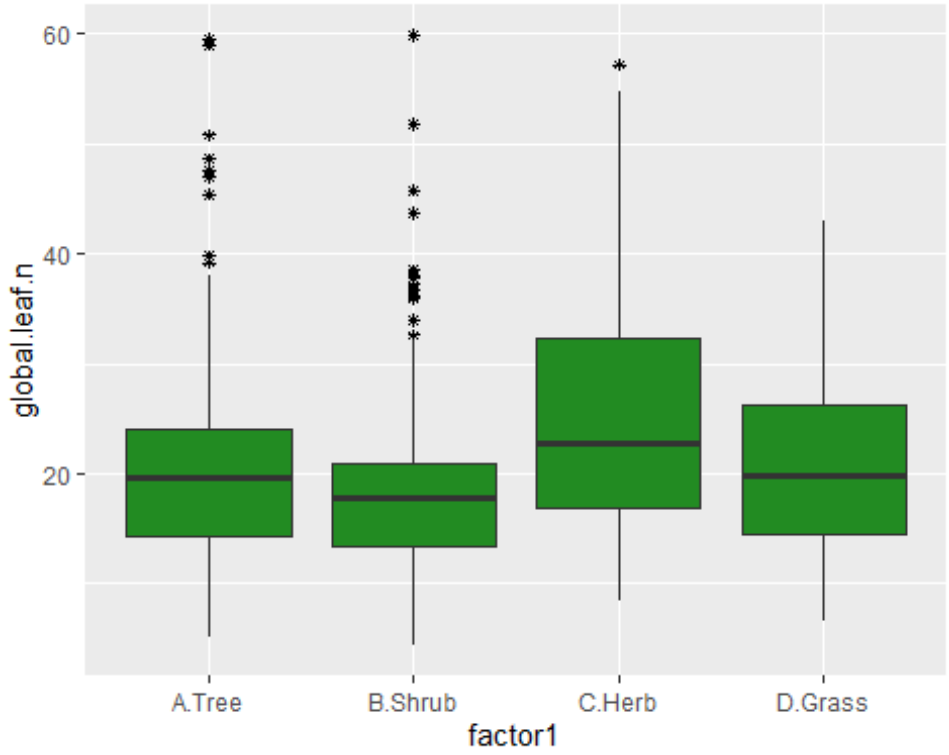
## Let's make the boxplot:

```
g <- ggplot(df) + # ggplot2 accessing our data
     geom_boxplot(aes(x=factor1, y=global.leaf.n))# making boxplot
                                                  # 'factor1' on x-axis
                                                 # 'global.leaf.n' on Y-axis
g
```



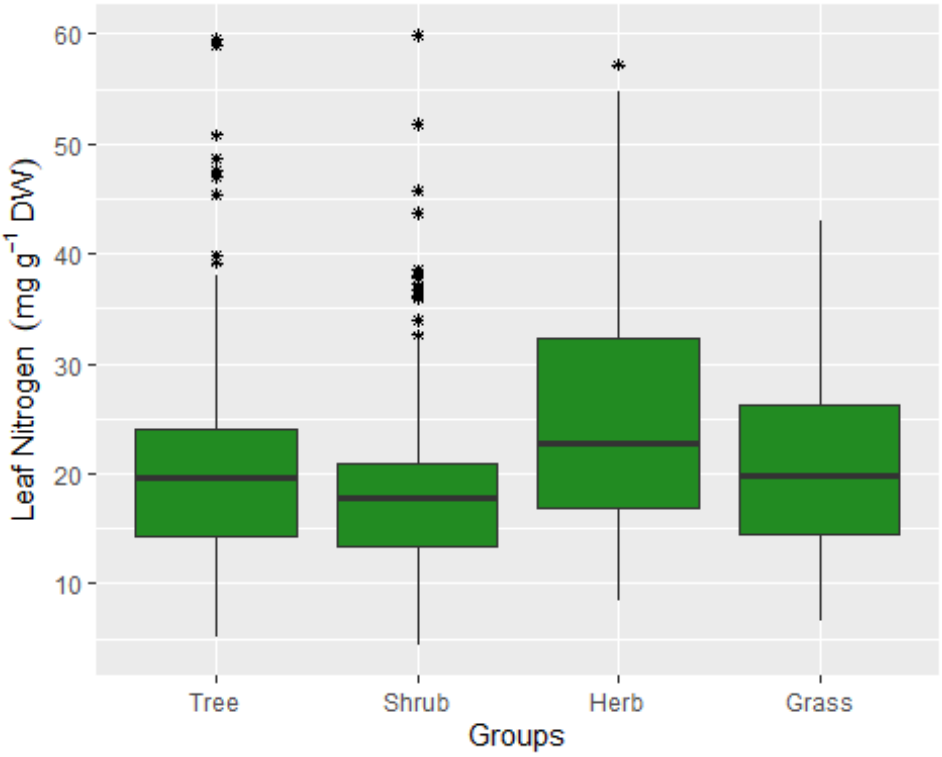

Well done! We've made our box plot. Now, we can include some additional commands to improve the appearance of the box plot.

```
g <- ggplot(df) +
      geom_boxplot(aes(x=factor1, y=global.leaf.n),
                   width=0.80, # we changed the width of the boxes
                   outlier.colour = "black", # we changed the outlier colour
                   outlier.shape = 8, # we changed the outlier shape
                   outlier.size = 1, # we changed the outlier size
                   fill="forestgreen", # we changed the box colour
                   lwd=0.6) # we changed the box lines' width
g
```



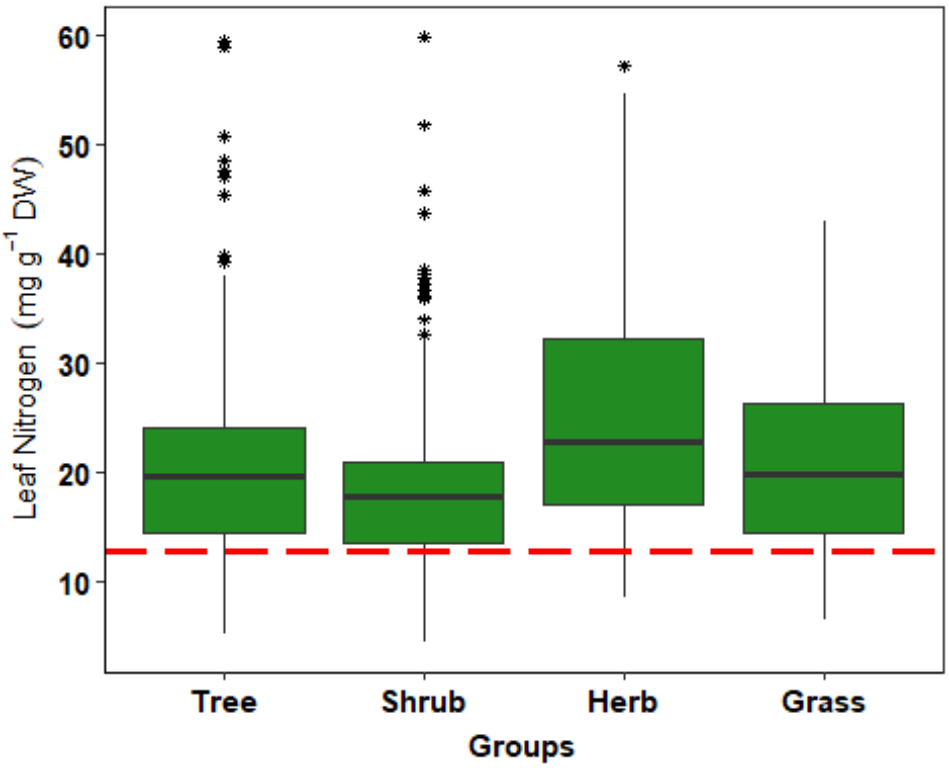

We can further improve the appearance ...

```
g <- ggplot(df) +
     geom_boxplot(aes(x=factor1, y=global.leaf.n),
                  width=0.80,
                  outlier.colour = "black",
                  outlier.shape = 8,
                  outlier.size = 1,
                  fill="forestgreen",
                  lwd=0.6)+

scale_x_discrete(labels=c("A.Tree"="Tree","B.Shrub"="Shrub","C.Herb"="Herb",
                          "D.Grass"="Grass","E.Conifer"="Conifer",
                          "F.Liana"="Liana"))+ # X-axis label
    scale_y_continuous(breaks=seq(0, 60, 10))+ # Y-axis scale changed
    ylab(expression('Leaf Nitrogen'~~(mg~g^-1~DW)))+ # Y-axis label changed
    xlab('Groups') # X-axis label changed
g
```




```
g <- ggplot(df) +
     geom_boxplot(aes(x=factor1, y=global.leaf.n),
                  width=0.80,
                  outlier.colour = "black",
                  outlier.shape = 8,
                  outlier.size = 1,
                  fill="forestgreen",
                  lwd=0.6)+

scale_x_discrete(labels=c("A.Tree"="Tree","B.Shrub"="Shrub","C.Herb"="Herb",
                          "D.Grass"="Grass","E.Conifer"="Conifer",
                          "F.Liana"="Liana"))+
    scale_y_continuous(breaks=seq(0, 60, 10))+
    ylab(expression('Leaf Nitrogen'~~(mg~g^-1~DW)))+
    xlab('Groups')+
    geom_hline(yintercept=12.7, col = "red", lty = 5, lwd = 1.2)+
                                                  # a red horizontal line
    theme_bw()+ # background colour changed
    theme(axis.text.x=element_text(color = 'black', size=11, face='bold'),
                                                  # X-axis text changed
          axis.title.x=element_text(color='black', size=11, face='bold',
                                    vjust =-0.35), # X-axis title changed
          axis.text.y=element_text(color = 'black', size=11,
                                   face='bold'), # Y-axis text changed
          axis.title.y=element_text(color='black', size=11,
                                    face='bold', vjust =1.5),
                                                  # Y-axis title changed
```

```
          panel.grid.major = element_blank(), # background major grid removed
          panel.grid.minor = element_blank(), # background minor grid removed
          panel.background = element_blank(),
          axis.line = element_line(colour = "black"))
                                              # background panel axis changed
g
```




```
global.leaf.n <- g # we named our box-plot as 'global.leaf.n'
```

**Saving the boxplot:**

```
dev.copy(png,'global.leaf.n.png',width = 13, height = 9, units="cm", res=400)
                    # we can change size and resolution of the plot
```

Congratulations! We just learned a coding skill on how to make a boxplot in R.