# An introduction to R

## Beginners session + Q&A

R-ladies London team | @RladiesLondon

26th October 2016

# What's R?

In origin was **S**, a programming language for statistical computing and interactive graphics. It was developed by John Chambers, Rick Becker and Allan Wilks of (NOKIA) Bell Laboratories in 1976.
S went through many version updates (1-4, 5 plus...) until in 1992 Ross Ihaka and Robert Gentleman (University of Auckland, New Zealand) worked on a further implementation and renamed it **R**.

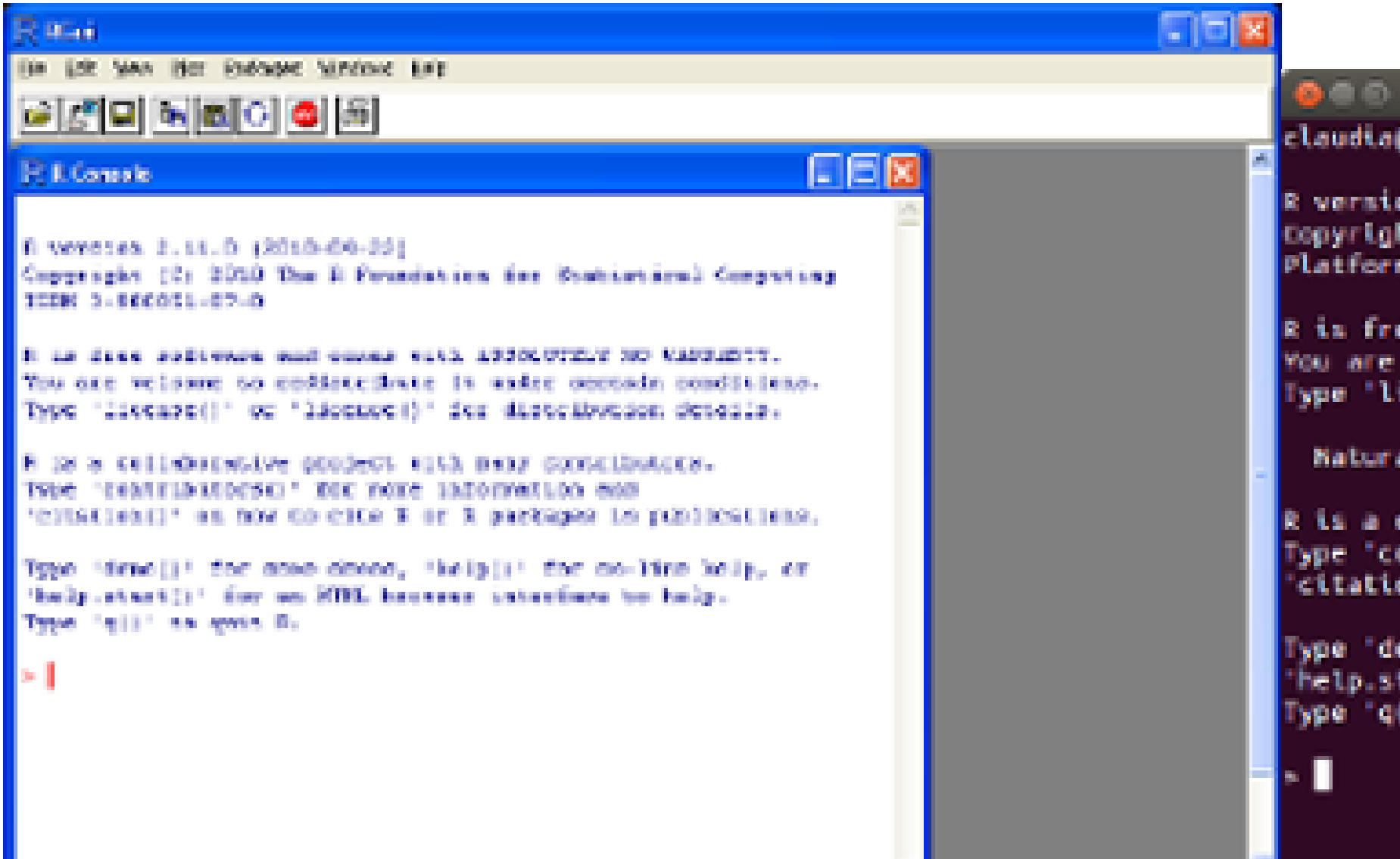
# Let's install the software

# Install R

Go to the R-project website, download and install R:

`https://www.r-project.org/`

The default Graphical User Interface (GUI) is basically a console!



# Your turn!

- Open an R instance

# Your turn!

- Open an R instance
- Top line in the opening message: **R version number (release date) and name**

# Your turn!

- Open an R instance
- Top line in the opening message: **R version number (release date) and name**
- Try the functions **license()**, **help()** and **q()**

# Your turn!

- Open an R instance
- Top line in the opening message: **R version number (release date) and name**
- Try the functions **license()**, **help()** and **q()**
- Use R as a basic calculator (+ - / *): 2 * 3

# Your turn!

- Open an R instance
- Top line in the opening message: **R version number (release date) and name**
- Try the functions **license()**, **help()** and **q()**
- Use R as a basic calculator (+ - / *): 2 * 3
- You can assign a value to a variable using either <- or =

# Your turn!

- Open an R instance
- Top line in the opening message: **R version number (release date) and name**
- Try the functions **license()**, **help()** and **q()**
- Use R as a basic calculator (+ - / *): 2 * 3
- You can assign a value to a variable using either <- or =
- Define variable 'a' equal to 1: a <- 1

# Your turn!

- Open an R instance
- Top line in the opening message: **R version number (release date) and name**
- Try the functions **license()**, **help()** and **q()**
- Use R as a basic calculator (+ - / *): 2 * 3
- You can assign a value to a variable using either <- or =
- Define variable 'a' equal to 1: a <- 1
- Define variable 'b' equal to 3 and calculate 'a+b': b <- 3; a+b

# Your turn!

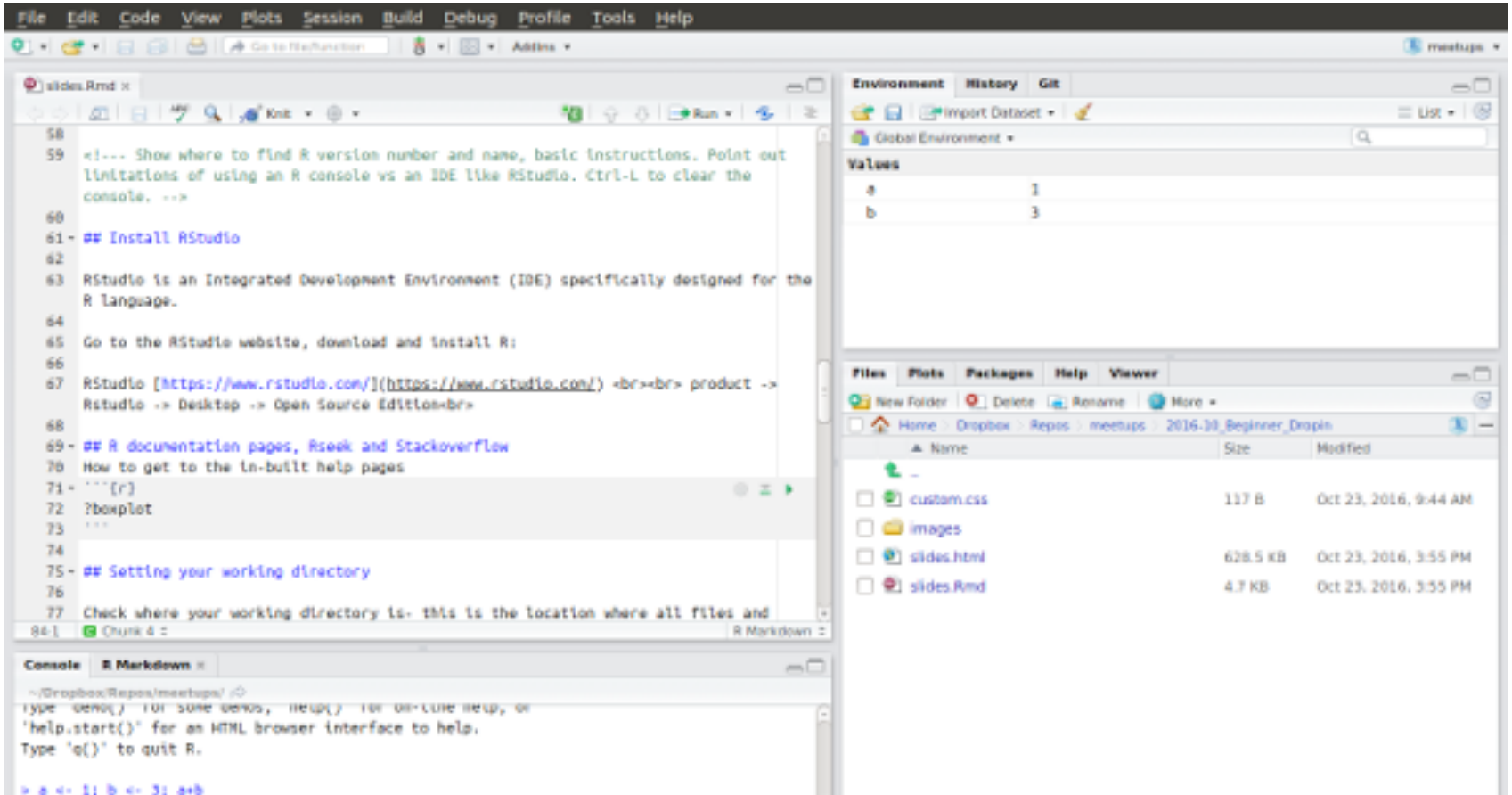
- Open an R instance
- Top line in the opening message: **R version number (release date) and name**
- Try the functions **license()**, **help()** and **q()**
- Use R as a basic calculator (+ - / *): 2 * 3
- You can assign a value to a variable using either <- or =
- Define variable 'a' equal to 1: a <- 1
- Define variable 'b' equal to 3 and calculate 'a+b': b <- 3; a+b
- Note the semicolon (;) is used to concatenate multiple operations in one line.

# Your turn!

- Open an R instance
- Top line in the opening message: **R version number (release date) and name**
- Try the functions **license()**, **help()** and **q()**
- Use R as a basic calculator (+ - / *): 2 * 3
- You can assign a value to a variable using either <- or =
- Define variable 'a' equal to 1: a <- 1
- Define variable 'b' equal to 3 and calculate 'a+b': b <- 3; a+b
- Note the semicolon (;) is used to concatenate multiple operations in one line.
- **Let's face it, using R from console is not very appealing! For complex operations you need to have an editor, check your environmental variables, visualise plots without switching window... that's why we use RStudio!**

# Install RStudio

RStudio is an Integrated Development Environment (IDE) specifically designed for the R language.
Go to the RStudio website, download and install R:
RStudio `https://www.rstudio.com/` product -> Rstudio -> Desktop -> Open Source Edition



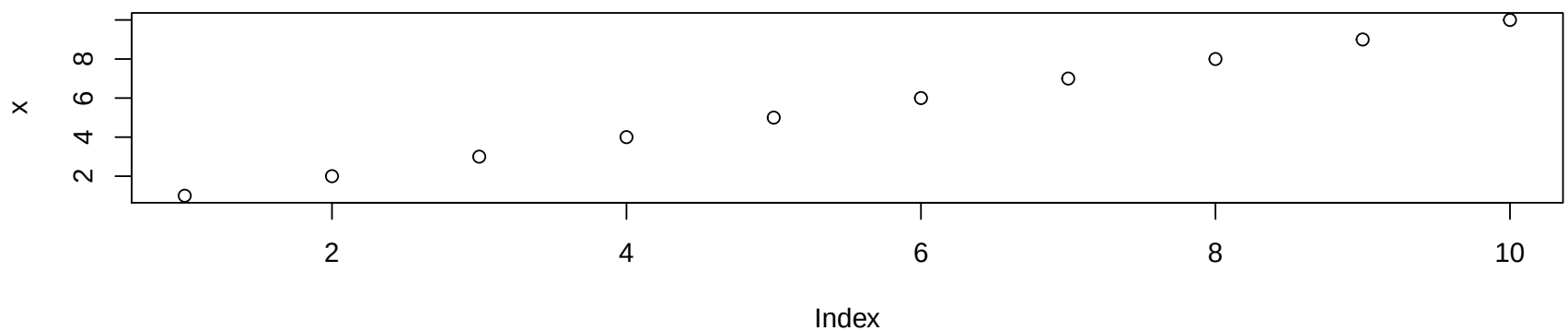
# Pre-installed packages

The basic R installation comes with a number of pre-installed packages (e.g. base, graphics, stats. . . ). A package is a container of functions, for instance the functions c() and mean() are in the **base** package, plot() is in the **graphics** package, ect.

```
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
mean(x)
```

```
## [1] 5.5
```

```
plot(x)
```



# Install additional packages

There are ~8000 R packages available on the Comprehensive R Archive Network (CRAN).

```r
# Install a new package for advanced graphics
install.packages("ggplot2")

# Load the package
library("ggplot2")
```

# Need help? R documentation pages, Rseek and Stackoverflow

How to get to the in-built help pages

```
?boxplot
```

Other ways to find help:

- Browse `http://rseek.org/` to find out what packages are available for a given topic (e.g. cluster analysis)

# Need help? R documentation pages, Rseek and Stackoverflow

How to get to the in-built help pages

```
?boxplot
```

Other ways to find help:

- Browse `http://rseek.org/` to find out what packages are available for a given topic (e.g. cluster analysis)
- If you get stuck, Google is your friend!

# Need help? R documentation pages, Rseek and Stackoverflow

How to get to the in-built help pages

```
?boxplot
```

Other ways to find help:

- Browse `http://rseek.org/` to find out what packages are available for a given topic (e.g. cluster analysis)
- If you get stuck, Google is your friend!
- If that doesn't work, post a question on `http://stackoverflow.com/`!

# Setting your working directory

Check where your working directory is- this is the location where all files and functions will be read and written to

```
getwd()
```

If this isn't where your files are then you can set it through (put your own path in the quotes!)

```
setwd("C:\Documents\mypath")
```

Or in Rstudio you can go: Session>Set Working Directory>Choose Directory and navigate to your folder

# Load & explore a dataset

Have a look at the in-built data sets in R

```
data()
```

# Look at the in-built data set on trees

```
trees
```

```
##    Girth Height Volume
## 1    8.3     70   10.3
## 2    8.6     65   10.3
## 3    8.8     63   10.2
## 4   10.5     72   16.4
## 5   10.7     81   18.8
## 6   10.8     83   19.7
## 7   11.0     66   15.6
## 8   11.0     75   18.2
## 9   11.1     80   22.6
## 10  11.2     75   19.9
## 11  11.3     79   24.2
## 12  11.4     76   21.0
## 13  11.4     76   21.4
## 14  11.7     69   21.3
## 15  12.0     75   19.1
```

# Explore the trees data set

The top of the data

```
head(trees)
```

The end of the data set

```
tail(trees)
```

The size and type of the data

```
str(trees)
```

Summary statistics on each of the fields

```
summary(trees)
```

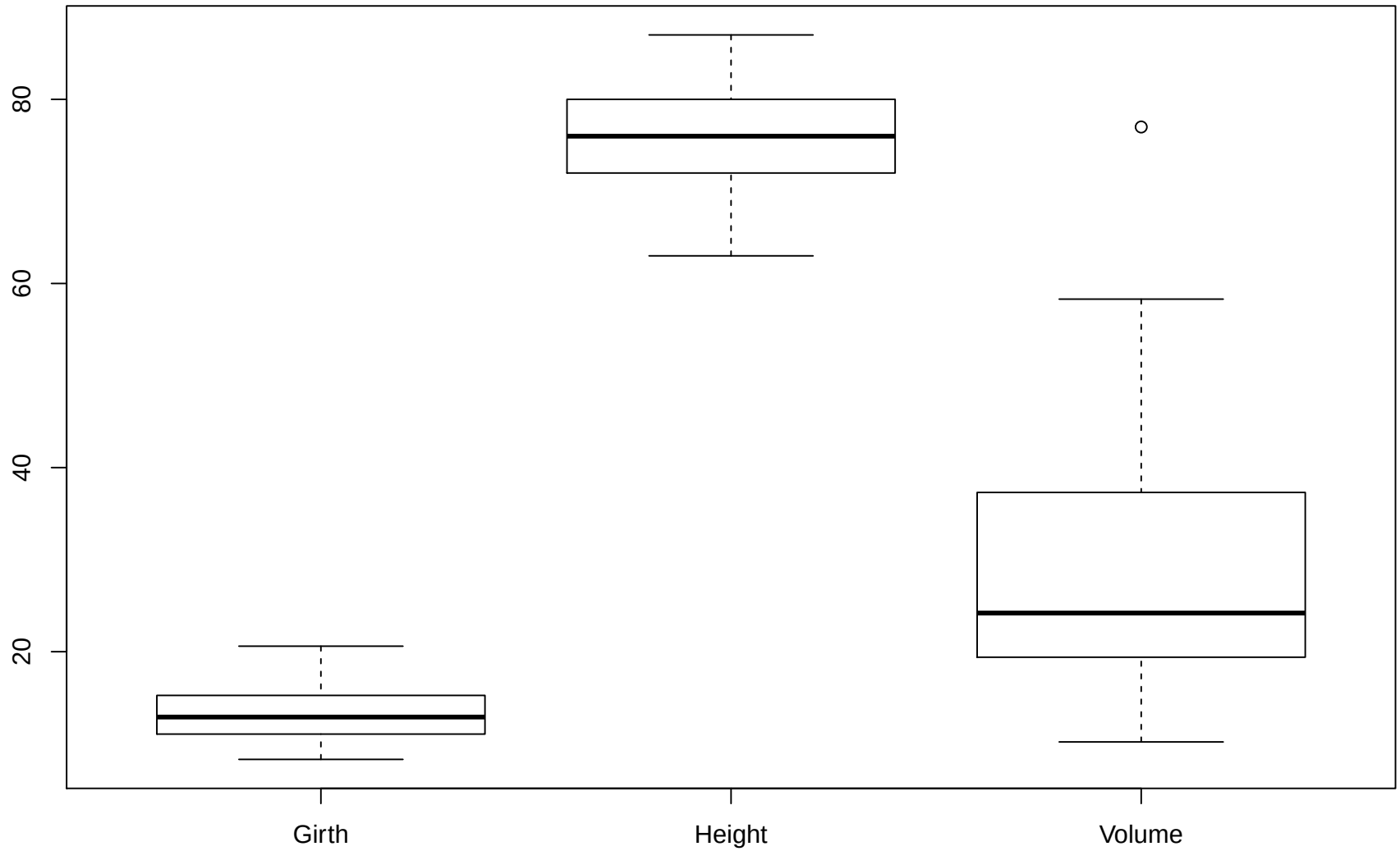
Pull out only one of the fields

```
summary(trees$Girth)
```

# Basic plots

## Create your first plot

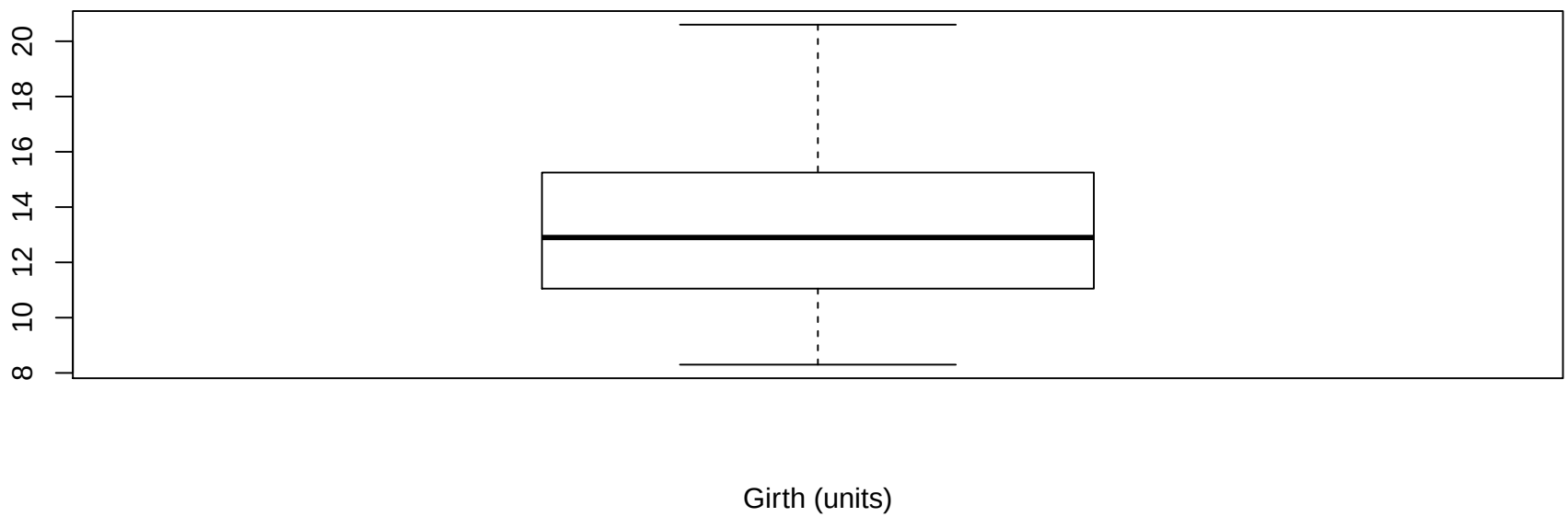
```
boxplot(trees)
```



# Basic plots

Create another variable with only one of the fields and plot that

```
treeGirth <- trees$Girth
boxplot(treeGirth, xlab="Girth (units)", title="Tree Girths
```



# Import data from a csv file

It would be best if you can get your own data and start exploring that. Make sure that your working directory is set to your file's location

```
data <- read.csv("myfile.csv")
```

# Where to go next

Great tutorials:

- edx MiT course: `https://www.edx.org/course/analytics-edge-mitx-15-071x-2`
- DataCamp: `https://www.datacamp.com/`
- Coursera: `https://www.coursera.org/learn/r-programming`
- Great Kaggle Tutorials: `https://www.kaggle.com/mrisdal/titanic/exploring-survival-on-the-titanic`