

# Introduction to R

## Session 1 – Introduction

Statistical Consulting Centre

consulting@stat.auckland.ac.nz  
The Department of Statistics  
The University of Auckland

19 July, 2017



**SCIENCE**  
DEPARTMENT OF STATISTICS

Each session comprises two parts: lecture and practice.

Session	Time	Session
1	09:00am - 10:30am	Introduction
	10:30am - 10:50am	Break
2	10:50am - 01:00pm	Subsetting data
	01:00pm - 02:00pm	Lunch break
3	02:00pm - 03:00pm	Data manipulation
	03:00pm - 03:20pm	Break
4	03:20pm - 04:30pm	Data exploration

Each session comprises two parts: lecture and practice.

Session	Time	Session
1	09:00am - 10:30am	Graphics
	10:30am - 10:50am	Break
2	10:50am - 01:00pm	Advanced Graphics
	01:00pm - 02:00pm	Lunch break
3	02:00pm - 03:00pm	Simple analysis
	03:00pm - 03:20pm	Break
4	03:20pm - 04:30pm	Advanced analysis

# R and UoA's Department of Statistics

- R was initially written by Robert Gentleman and Ross Ihaka *R & R* of the **Department of Statistics, University of Auckland**.
- Three members of the *R Development Core Team* are in UoA's Department of Statistics.



**SCIENCE**  
DEPARTMENT OF STATISTICS



Ross Ihaka



Robert Gentleman



Paul Murrell



Thomas Lumley

# R and UoA's Department of Statistics

What does this mean?

*If you want to learn R, you are talking to the right people!*



**Chris Triggs**

Director Consulting Services  
Phone: +64 9 373 7599 ext 88856  
Email: [triggs@stat.auckland.ac.nz](mailto:triggs@stat.auckland.ac.nz)

For more information, please see Chris's profile.



**Yannan Jiang**

Senior Research Fellow  
Phone: +64 9 373 7599 ext 84725  
Email: [y.jiang@auckland.ac.nz](mailto:y.jiang@auckland.ac.nz)

For more information, please see Yannan's profile.



**Kathy Ruggiero**

Senior Lecturer  
Phone: +64 9 373 7599 ext 89456  
Email: [k.ruggiero@auckland.ac.nz](mailto:k.ruggiero@auckland.ac.nz)

For more information, please see Kathy's profile.



**Jessica McLay**

Research Fellow  
Phone: +64 9 373 7599 ext 73678 or 85313  
Email: [jessica.mclay@auckland.ac.nz](mailto:jessica.mclay@auckland.ac.nz)

For more information, please see Jessica's profile.



**Rachel Chen**

Research Fellow  
Phone: +64 9 373 7599 ext 89384  
Email: [rachel.chen@auckland.ac.nz](mailto:rachel.chen@auckland.ac.nz)

For more information, please see Rachel's profile.



**Avinesh Pillai**

Research Fellow  
Phone: +64 9 373 7599 ext 82368 (Mon-Wed) or ext 81169 (Thurs & Fri)  
Email: [a.pillai@auckland.ac.nz](mailto:a.pillai@auckland.ac.nz)

For more information, please see Avinesh's profile.

# What is 'R'?

## What does this mean?

R is a free software environment for statistical computing and graphics"

Key words:

- FREE!
- Statistical computing
- Graphics (much more flexible than SAS, SPSS, JMP, etc.)
- Support from communities of different fields, i.e. R packages.  
<https://cran.r-project.org/web/views/> and Bioconductor  
<https://www.bioconductor.org/>.
- Even Microsoft is in it: Microsoft R Open.  
<https://mran.microsoft.com/open/>.



# What is 'R'?

## What does this mean?

R is a free software environment for statistical computing and graphics"

R and the biological sciences:

- Many *applications of statistical methods to biological datasets are implemented in R*
- These R *packages* are publically available on the web for immediate download and use.
- E.g. Next Generation Sequencing, Genomics (Bioconductor).

# How to download and install R

- ➊ Go to the CRAN (Comprehensive R Archive Network)  
`cran.stat.auckland.ac.nz`.
- ➋ Download the relevant version for Linux/Mac/Windows.
  - We will only look at R in the Windows environment today.
- ➌ Install it on your computer (for Windows only):
  - Choose “Yes (customized startup)” in Startup options.
  - Choose “SDI (separate windows)” in Display mode.
  - Choose “HTML help” in Help .

# Using the R editor

- The R GUI is not menu driven.
- Commands can be typed at the console.
  - OK for simple calculations requiring few lines of code
  - Painful for anything more!
- We *strongly* recommend using an R editor
  - Great for reproducible analyses and research.
  - Best editor for you depends on whether you are a(n)...
  - ① Beginner: Built-in R editor,
  - ② Advanced user: Rstudio, Tinn-R, Notepad++, and many others.
  - ③ R geek: Emacs

- Helps in write better R code.
- Produce reports (Rmarkdown).
- Produce interactive reports/tools (Shiny).
- Develop R packages.

# Using R as a calculator

```
1 + 2
```

```
## [1] 3
```

```
1 + 3^2
```

```
## [1] 10
```

```
log(15) - sqrt(3.4)
```

```
## [1] 0.8641413
```

```
pnorm(1.96)
```

```
## [1] 0.9750021
```

# Variable assignment

- `<-` is the “assign to” operator, made up of `<` and `-` without a space.
- E.g., `x <- 2` is read as “The value 2 is assigned to the object x”.

```
x <- 2  
y <- 3  
x^2 - 3 * y + 5
```

```
## [1] 0
```

- `<-` has a direction, from right to left, `x <- 2` means assigning 2 to x,

# Variable assignment

- `->` operates from left to right, assigning `x` to `2`.
  - `2` is a real value so you can not do that.

```
2 <- x
```

```
## Error in 2 <- x: invalid (do_set) left-hand side to assign
```

- `=` has no direction and can be confusing sometimes.
- It is good programming practice to use `<-`.
- The most important thing is to keep consistent.

# Getting help

- Google!!!!  
e.g. How to calculate the mean in R? The search results tell you that the function `mean()` would be helpful.
- Quick-R: <http://www.statmethods.net/>
- R-bloggers: <https://www.r-bloggers.com/>



# Getting help

- `?`  
e.g. `?mean` brings up the help file for this function. It will tell you (almost) everything you need to know to use `mean()`.
- `??`  
e.g. `??mean` searches for everything related to `mean` in your computer.
- `RSiteSearch(" ")`  
Searches everything on CRAN as well as your computer.

- Statisticians prefer (read: **want**) rectangular data files
  - Each case in its own row
  - Data collected on each variable in its own column
  - Variable names in the first row of each column
  - No blanks, e.g. fill with NA, \*, 99999, anything but a blank!
- R likes (read: **needs**) this too!
- R prefers to read data files in Comma Separated Value (CSV) format.
- This does not mean R only reads files stored in csv format.

# Getting data into R

Try your best to save your data in a csv or txt format.

- Most datasets are saved in an Excel spreadsheet.
- Do as much data cleaning as you can in Excel. No comments, no formatting, no colours, no fancy fonts.
- Convert it into csv by clicking on Save As. Change the Save as type from `xlsx` or `xls` into CSV (Comma Delimited).
- CSV can have one worksheet only. If you have multiple worksheets, it saves the active worksheet.

# Read and Check

- Always set a working directory using `setwd()`, this can be a directory where you store the data and/or outputting the results.
- Use `read.csv` to read a CSV file into R.
- `dim()`: Returns the number of observations (rows) and variables (columns).
- `head()/tail()`: Returns the first/last few rows of a data set.
- `str()`: Returns the structure of the dataset, e.g., dimension, column names, type of data object, first few values of each variable.
- `names()`: Returns the names of the variables contained in a dataset.

Five variables:

- C02: current or double (the current) CO<sub>2</sub> level.
- Species: *Psidium guajava* (PG), *Archontophoenix cunninghamiana* (AC) and *Scheffera actinophylla* (SA).
- root: root biomass
- shoot: shoot biomass
- biomass: total biomass

# Reading data into R

```
setwd("your working directory")  
Growth.df <- read.csv("Growth.csv")  
head(Growth.df)
```

##		C02 Species	root	shoot	biomass
## 1	current	SA	2.0203	6.8292	8.8495
## 2	current	SA	1.0681	5.2047	6.2728
## 3	current	SA	2.0499	NA	9.3255
## 4	current	SA	2.6797	5.6128	8.2925
## 5	current	AC	0.5098	1.8772	2.3870
## 6	current	AC	1.0511	4.1917	5.2428

## dim() and str()

```
dim(Growth.df)
str(Growth.df)
```

```
## [1] 144    5
```

```
## 'data.frame':    144 obs. of  5 variables:
```

```
## $ C02      : Factor w/ 2 levels "current","double": 1 1 1 1
```

```
## $ Species: Factor w/ 3 levels "AC","PG","SA": 3 3 3 3 1 1
```

```
## $ root     : num  2.02 1.07 2.05 2.68 0.51 ...
```

```
## $ shoot    : num  6.83 5.2 NA 5.61 1.88 ...
```

```
## $ biomass: num  8.85 6.27 9.33 8.29 2.39 ...
```

## names(Growth.df)

```
# Names of the variables  
names(Growth.df)
```

```
## [1] "CO2"      "Species" "root"     "shoot"    "biomass"
```

- Anything following the # symbol is treated as a comment and ignored by R.
- Writing comments is a very good habit to develop!



# Descriptive statistics

Calculate the mean of biomass:

```
mean(biomass)
```

```
## Error in mean(biomass): object 'biomass' not found
```

You must tell R that biomass is a variable (column) *within* Growth.df, i.e.

```
mean(Growth.df$biomass)
```

```
## [1] NA
```

You must also tell R how to deal with missing values: remove them before calculating the mean, i.e.

```
mean(Growth.df$biomass, na.rm = TRUE)
```

```
## [1] 7.569813
```

# table of counts

```
# One-way table of counts
```

```
table(Growth.df$Species)
```

```
##
```

```
## AC PG SA
```

```
## 48 48 48
```

# table of proportions

```
# Total count
```

```
total <- sum(table(Growth.df$Species))  
total
```

```
## [1] 144
```

```
# Proportions of total
```

```
table(Growth.df$Species)/total
```

```
##
```

```
##           AC           PG           SA
```

```
## 0.3333333 0.3333333 0.3333333
```

# One-way tables with less typing

Tired of typing `Growth.df$` over and over again? Use the `with` function.

```
Species.table <- with(Growth.df, table(Species))
Species.table
```

```
## Species
## AC PG SA
## 48 48 48
```

```
total <- sum(Species.table)
Species.table/total
```

```
## Species
##          AC          PG          SA
## 0.3333333 0.3333333 0.3333333
```

# One-way tables with less typing

```
# Convert to percentages
```

```
Species.pct <- 100 * Species.table/total  
Species.pct
```

```
## Species
```

```
##          AC          PG          SA  
## 33.33333 33.33333 33.33333
```

```
# Round to 1 decimal place
```

```
round(Species.pct, 1)
```

```
## Species
```

```
##      AC      PG      SA  
## 33.3 33.3 33.3
```

# Two-way frequency tables

```
Species.CO2.tab <- with(Growth.df, table(Species, CO2))  
Species.CO2.tab
```

```
##           CO2  
## Species current double  
##      AC      24      24  
##      PG      24      24  
##      SA      24      24
```

# Two-way frequency tables

```
# Calculate proportion with respect to 'margin' total margin =  
# or 2 (column total)  
perc.Species.CO2 <- prop.table(Species.CO2.tab, margin = 2)  
perc.Species.CO2
```

```
##           CO2  
## Species  current    double  
##      AC 0.3333333 0.3333333  
##      PG 0.3333333 0.3333333  
##      SA 0.3333333 0.3333333
```

# Two-way frequency tables

```
# Tabulate as percentages  
round(100 * perc.Species.CO2, 1)
```

```
##           CO2  
## Species current double  
##      AC      33.3    33.3  
##      PG      33.3    33.3  
##      SA      33.3    33.3
```



# Summary

- Quick introduction to R
- Getting data into R
- Frequency tables