Introduction to R

Marcus W. Beck   Sergey Berg

Department of Fisheries, Wildlife, and Conservation Biology
University of Minnesota, Twin Cities

May 21, 2013

What you'll learn about

- What is R?
- What's possible with R?
  - CRAN and packages
- R basics
  - Installation
  - Command-line interface
  - Coding basics
  - Functions and objects
  - Data import and manipulation
- Help!

What you'll learn about

- What is R?
- What's possible with R?
  - CRAN and packages
- R basics
  - Installation
  - Command-line interface
  - Coding basics
  - Functions and objects
  - Data import and manipulation
- Help!

Interactive!

What is R ?

R is a language and environment for statistical computing and graphics [r-project.org]

What is R ?

R is a language and environment for statistical computing and graphics [r-project.org]

R is a computer language that allows the user to program algorithms and use tools that have been programmed by others [Zuur et al. 2009]

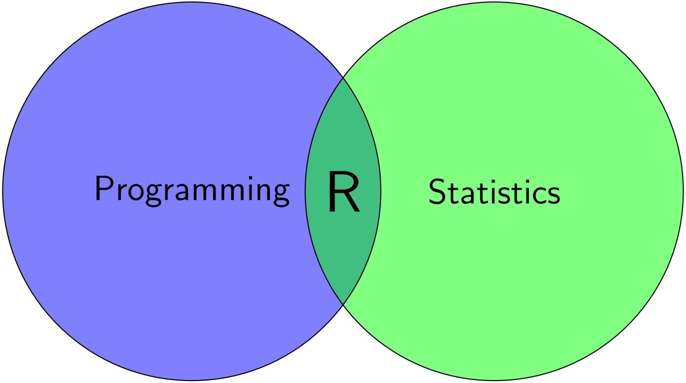
What is R ?

R is a language and environment for statistical computing and graphics [r-project.org]

R is a computer language that allows the user to program algorithms and use tools that have been programmed by others [Zuur et al. 2009]

Different from other statistics software because it is also a programming language...

What is R ?



Programming

R

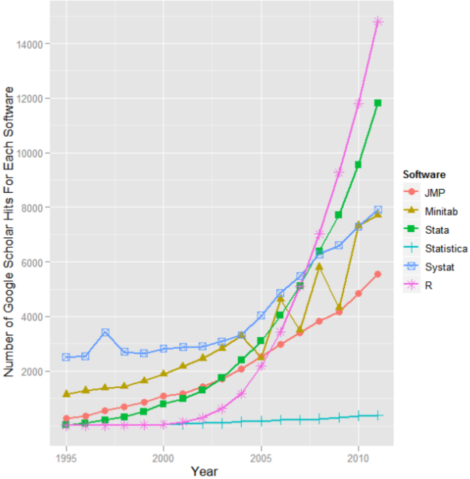
Statistics

R is both... this creates a steep learning curve.

What is R ?

R is becoming the statistical software of choice

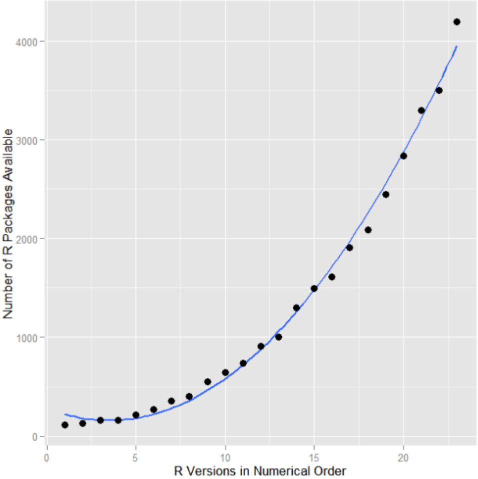
Plot of Google scholar hits over time for different software packages [r4stats.com]



What is R ?

R is becoming the statistical software of choice

Exponential growth in number of contributed packages [r4stats.com]



What's possible with R ?

R is incredibly flexible, if you want something done, someone else has written the code...

What's possible with R ?

R is incredibly flexible, if you want something done, someone else has written the code...

R is open-source software, which mean it's free and is supported by a large network of contributors - the Comprehensive R Network [CRAN]

What's possible with R ?

R is incredibly flexible, if you want something done, someone else has written the code...

R is open-source software, which mean it's free and is supported by a large network of contributors - the Comprehensive R Network [CRAN]

CRAN is a collection of sites which carry identical material, consisting of the R distribution(s), the contributed extensions, documentation for R, and binaries [R FAQ]

What's possible with R ?

R is incredibly flexible, if you want something done, someone else has written the code...

R is open-source software, which mean it's free and is supported by a large network of contributors - the Comprehensive R Network [CRAN]

CRAN is a collection of sites which carry identical material, consisting of the R distribution(s), the contributed extensions, documentation for R, and binaries [R FAQ]

Basically a repository of R utilities that others have written

What's possible with R ?

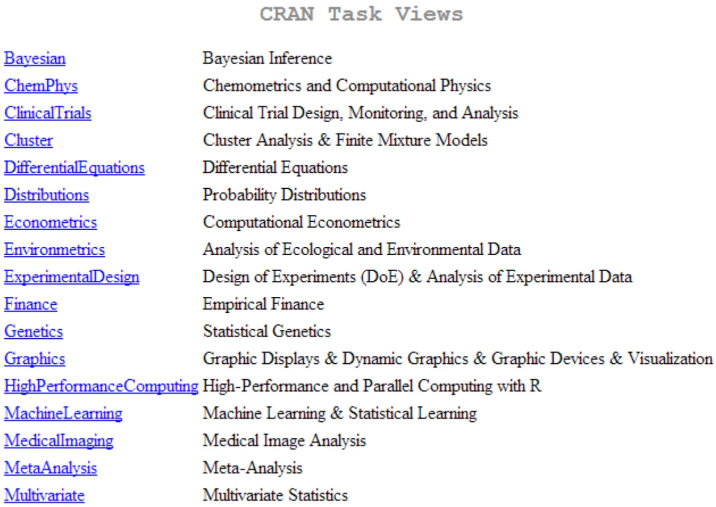
R is incredibly flexible, if you want something done, someone else has written the code...

R is open-source software, which mean it's free and is supported by a large network of contributors - the Comprehensive R Network [CRAN]

CRAN is a collection of sites which carry identical material, consisting of the R distribution(s), the contributed extensions, documentation for R, and binaries [R FAQ]

Basically a repository of R utilities that others have written - the CRAN task views contain descriptions of contributed packages by category

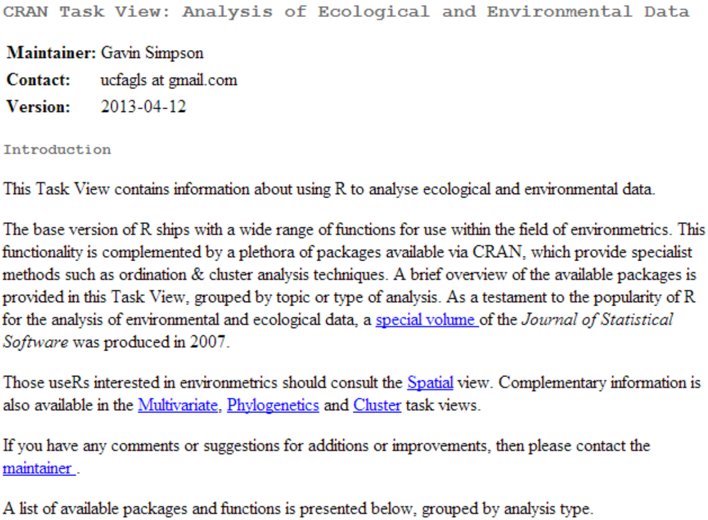
What's possible with R ?



CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
Finance	Empirical Finance
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
MetaAnalysis	Meta-Analysis
Multivariate	Multivariate Statistics

What's possible with R ?



CRAN Task View: Analysis of Ecological and Environmental Data

Maintainer: Gavin Simpson
Contact: ucfagls at gmail.com
Version: 2013-04-12

Introduction

This Task View contains information about using R to analyse ecological and environmental data.

The base version of R ships with a wide range of functions for use within the field of environmetrics. This functionality is complemented by a plethora of packages available via CRAN, which provide specialist methods such as ordination & cluster analysis techniques. A brief overview of the available packages is provided in this Task View, grouped by topic or type of analysis. As a testament to the popularity of R for the analysis of environmental and ecological data, a special volume of the *Journal of Statistical Software* was produced in 2007.

Those useRs interested in environmetrics should consult the Spatial view. Complementary information is also available in the Multivariate, Phylogenetics and Cluster task views.

If you have any comments or suggestions for additions or improvements, then please contact the maintainer .

A list of available packages and functions is presented below, grouped by analysis type.

What's possible with R ?

R has a base package that is included in installation, others are downloaded as needed

```
> install.packages('newpackage')
```

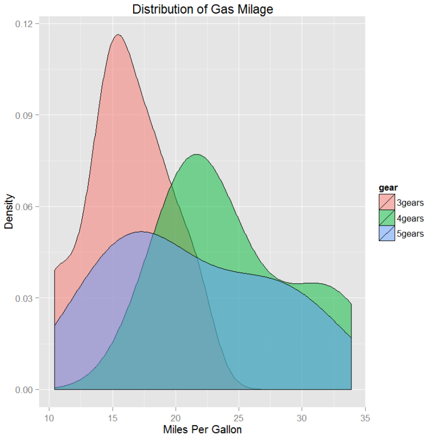
The base package will be sufficient for most of your needs - includes arithmetic, input/output, basic programming support, graphics, etc.

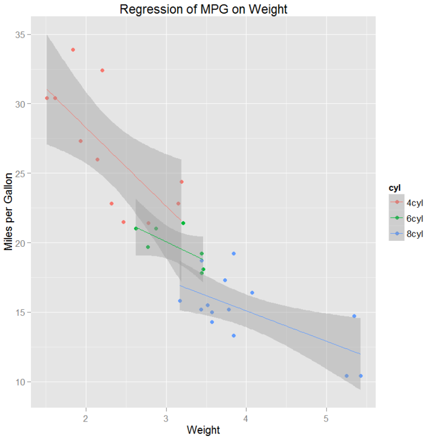
Contributed packages will meet your other needs - now exceed 4000

What's possible with R ?

```
> demo(package = .packages(all.available = TRUE))
```

List of demonstrations with available packages - examples from ggplot2 package

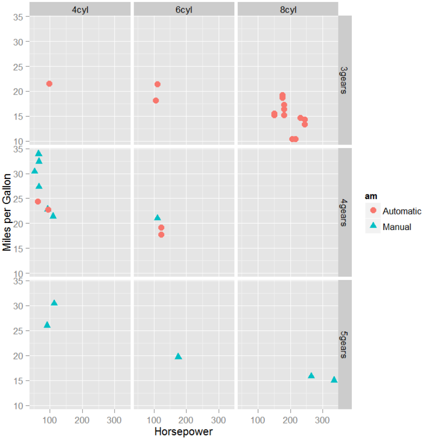


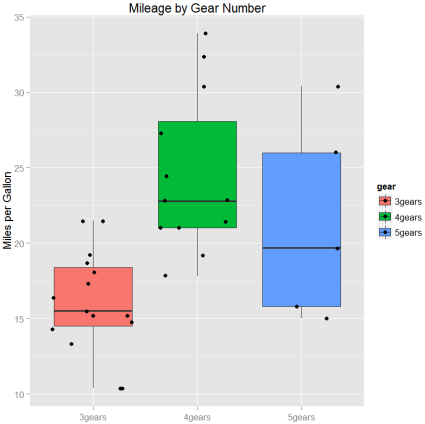


What's possible with R ?

```
> demo(package = .packages(all.available = TRUE))
```

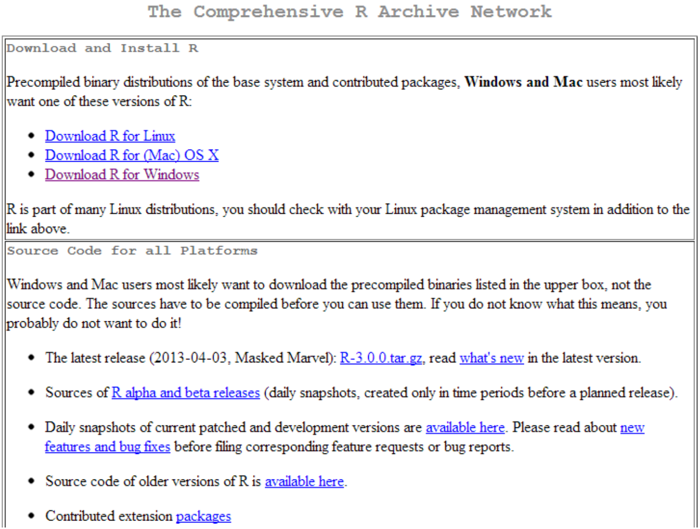
List of demonstrations with available packages - examples from ggplot2 package





R basics

Installation - visit r-project.org and follow directions



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

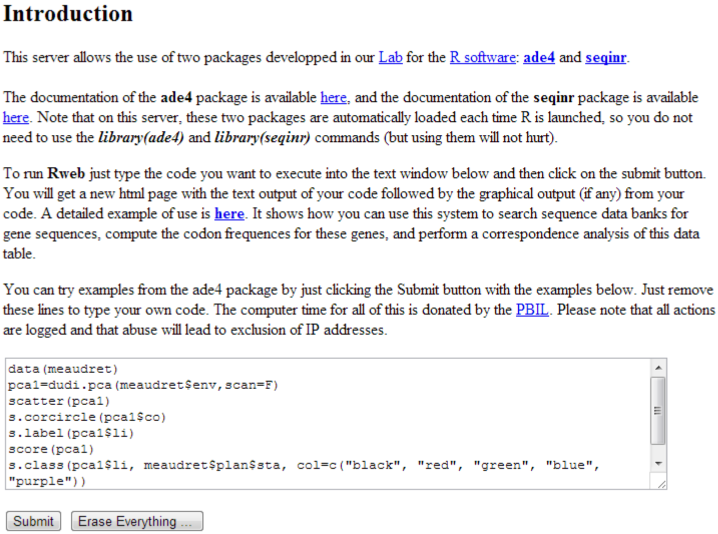
Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2013-04-03, Masked Marvel): R-3.0.0.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

basics

Or visit Rweb for an online version (not recommended)



Introduction

This server allows the use of two packages developped in our Lab for the R software: **ade4** and **seqinr**.

The documentation of the **ade4** package is available here, and the documentation of the **seqinr** package is available here. Note that on this server, these two packages are automatically loaded each time R is launched, so you do not need to use the ***library(ade4)*** and ***library(seqinr)*** commands (but using them will not hurt).

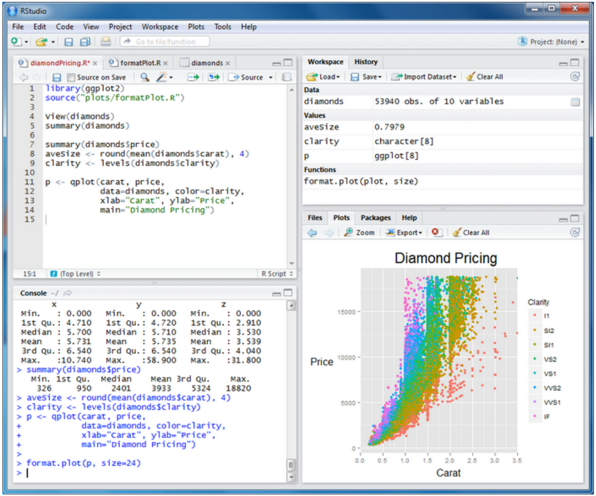
To run **Rweb** just type the code you want to execute into the text window below and then click on the submit button. You will get a new html page with the text output of your code followed by the graphical output (if any) from your code. A detailed example of use is **here**. It shows how you can use this system to search sequence data banks for gene sequences, compute the codon frequences for these genes, and perform a correspondence analysis of this data table.

You can try examples from the ade4 package by just clicking the Submit button with the examples below. Just remove these lines to type your own code. The computer time for all of this is donated by the PBIL. Please note that all actions are logged and that abuse will lead to exclusion of IP addresses.

```
data(meaudret)
pca1=dudi.pca(meaudret$env,scan=F)
scatter(pca1)
s.corcircle(pca1$co)
s.label(pca1$li)
score(pca1)
s.class(pca1$li, meaudret$plan$sta, col=c("black", "red", "green", "blue",
"purple"))
```

basics

A text editor is highly recommended, e.g. RStudio



```
library(ggplot2)
source("plots/formatPlot.R")

view(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")

```

```
       x                y                z         
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910  
 Median : 5.700   Median : 5.710   Median : 3.530  
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539  
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040  
 Max.   :10.740   Max.   :58.900   Max.   :31.800  
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    326     950    2401    3933    5324   18820 
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
> 
> format.plot(p, size=24)
> 
```

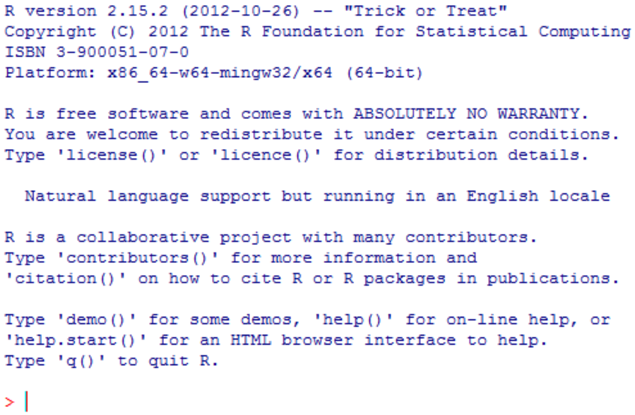
```
Data
diamonds      53940 obs. of 10 variables
Values
aveSize       0.7979
clarity       character[8]
p             ggplot[8]
Functions
format.plot(plot, size)
```

R basics

How is R different from Excel?

basics

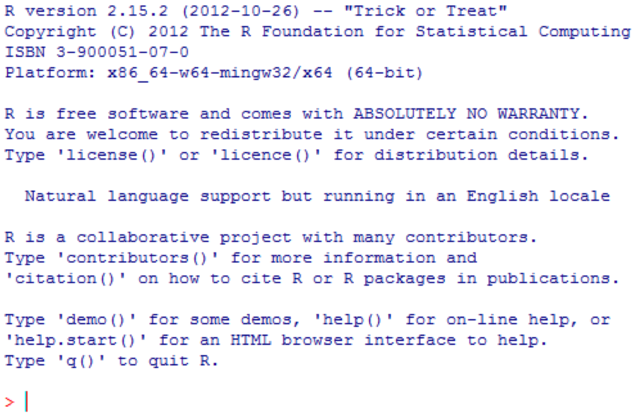
How is R different from Excel? R is a command-line interface



```
R version 2.15.2 (2012-10-26) -- "Trick or Treat"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

R basics

How is R different from Excel? R is a command-line interface



```
R version 2.15.2 (2012-10-26) -- "Trick or Treat"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

What next??

R basics

Lines of code are executed by R at the prompt (>)

R basics

Lines of code are executed by R at the prompt (>)

Enter the code and press enter, the output is returned

```
> print('hello world!')
[1] "hello world!"
> 2+2
[1] 4
> (2+2)/4
[1] 1
> rep("a",4)
[1] "a" "a" "a" "a"
```

R basics

A disadvantage of code is that everything entered must be 100 % correct

```
> 2+2a

Error: unexpected symbol in "2+2a"

> a

Error: object 'a' not found
```

R basics

A disadvantage of code is that everything entered must be 100 % correct

```
> 2+2a
Error: unexpected symbol in "2+2a"
> a
Error: object 'a' not found
```

But this enables a complete documentation of your workflow...

...your code is a living document of your analyses.

R basics

Assigning data to R objects is critical for analysis

basics

Assigning data to R objects is critical for analysis

Assignment is possible using <- or =

```
> a<-1
> 2+a
[1] 3
> a=1
> 2+a
[1] 3
> a=2+2
> a/4
[1] 1
```

R basics

Assigning data to R objects is critical for analysis

More complex assignments are possible

```
> a<-c(1,2,3,4)
> a

[1] 1 2 3 4

> a<-seq(1,4)
> a

[1] 1 2 3 4

> a<-c("a","b","c")
> a

[1] "a" "b" "c"
```

basics

Anatomy of a function - functions perform tasks for you, much like in Excel

function(arguments)

R basics

Anatomy of a function - functions perform tasks for you, much like in Excel

function(arguments)

```
> c(1,2) #concatenate function

[1] 1 2

> mean(c(1,2)) #mean function

[1] 1.5

> seq(1,4) #create a sequence of values

[1] 1 2 3 4
```

R basics

Understanding classes of R objects is necessary for analysis

An object is any variable of interest that you want to work with

The class defines the type of information the object contains

R basics

Understanding classes of R objects is necessary for analysis

An object is any variable of interest that you want to work with

The class defines the type of information the object contains

Most common are ‘numeric’ or ‘character’ classes

```
> class(1)
[1] "numeric"
> class("1")
[1] "character"
```

‘Factors’ are also common, define categorical variables

R basics

Understanding classes of R objects is necessary for analysis

The classes of an object defines a protocol for evaluating or organizing variables

For example, we cannot add add two objects with different classes:

```
> '1' + 1

Error in "1" + 1 : non-numeric argument to binary operator
```

R basics

Objects (and their classes) can be stored in the computer's memory in different ways - aka the workspace for your R session

Most common structures are 'vectors' and 'data.frames'

R basics

Objects (and their classes) can be stored in the computer's memory in different ways - aka the workspace for your R session

Most common structures are 'vectors' and 'data.frames'

Vectors are a collection of objects of the same class (e.g., a column in a table), whereas a data frame is analogous to a table with rows and columns (e.g., collection of vectors)

R basics

Objects (and their classes) can be stored in the computer's memory in different ways - aka the workspace for your R session

Most common structures are 'vectors' and 'data.frames'

Vectors are a collection of objects of the same class (e.g., a column in a table), whereas a data frame is analogous to a table with rows and columns (e.g., collection of vectors)

```
> a<-c(1,2)
> a

[1] 1 2

> b<-c("a","b")
> b

[1] "a" "b"
```

R basics

Objects (and their classes) can be stored in the computer's memory in different ways - aka the workspace for your R session

Most common structures are ‘vectors’ and ‘data.frames’

Vectors are a collection of objects of the same class (e.g., a column in a table), whereas a data frame is analogous to a table with rows and columns (e.g., collection of vectors)

```
> a<-c(1,2)
> a

[1] 1 2

> b<-c("a","b")
> b

[1] "a" "b"
```

```
> c<-data.frame(a,b)
> c

  a b
1 1 a
2 2 b
```

basics

How are data imported into R?

R needs to know where the data are located on your computer:

```
> setwd("C:/projects/my_data/")
```

This establishes a 'working directory' for data import/export

R basics

How are data imported into R?

R needs to know where the data are located on your computer:

```
> setwd("C:/projects/my_data/")
```

This establishes a ‘working directory’ for data import/export

R can import almost any type of data but ‘spreadsheet’ or text-based files are most common

R basics

How are data imported into R?

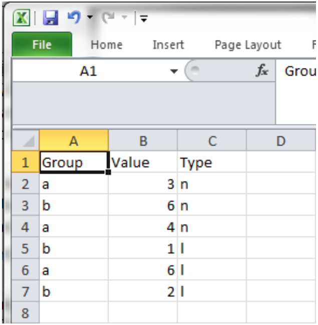
R can import Excel data using the RODBC package, but this is not simple

R basics

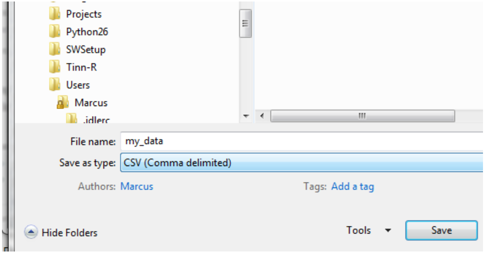
How are data imported into R?

R can import Excel data using the RODBC package, but this is not simple

The easiest approach is to format data in Excel then export to a .csv or .txt file



	A	B	C	D
1	Group	Value	Type	
2	a	3	n	
3	b	6	n	
4	a	4	n	
5	b	1	l	
6	a	6	l	
7	b	2	l	
8				



File name: my_data

Save as type: CSV (Comma delimited)

Authors: Marcus

Tags: Add a tag

R basics

How are data imported into R?

Use the read.table or read.csv functions to import the data, must be in your working directory

R basics

How are data imported into R?

Use the read.table or read.csv functions to import the data, must be in your working directory

```
> dat<-read.csv("my_data.csv",header=T)
> dat
```

```
  Group Value Type
1     a     3    n
2     b     6    n
3     a     4    n
4     b     1    l
5     a     6    l
6     b     2    l
```

R basics

How are data imported into R?

Use the read.table or read.csv functions to import the data, must be in your working directory

```
> dat<-read.table("my_data.csv",sep=',',header=T)
> dat

  Group Value Type
1     a     3    n
2     b     6    n
3     a     4    n
4     b     1    l
5     a     6    l
6     b     2    l
```

R basics

Imported data can be viewed several ways, view the whole object or parts

Rows or columns can be obtained by indexing with brackets separated by a comma: data[row,column]

R basics

Imported data can be viewed several ways, view the whole object or parts

Rows or columns can be obtained by indexing with brackets separated by a comma: data[row,column]

```
> dat
  Group Value Type
1     a     3    n
2     b     6    n
3     a     4    n
4     b     1    l
5     a     6    l
6     b     2    l
```

R basics

Imported data can be viewed several ways, view the whole object or parts

Rows or columns can be obtained by indexing with brackets separated by a comma: data[row,column]

```
> dat
  Group Value Type
1     a     3    n
2     b     6    n
3     a     4    n
4     b     1    l
5     a     6    l
6     b     2    l
```

```
> dat[1,] #row 1
  Group Value Type
1     a     3    n
> dat[,2] #column 2
[1] 3 6 4 1 6 2
> dat[4,1] #row 4, column 1
[1] b
Levels: a b
```

basics

Imported data can be viewed several ways, view the whole object or parts

Access using column names or the attach function

```
> dat$Value

[1] 3 6 4 1 6 2

> dat[,'Value']

[1] 3 6 4 1 6 2
```

```
> attach(dat)
> Value

[1] 3 6 4 1 6 2
```

R basics

Imported data can be viewed several ways, view the whole object or parts

Access using column names or the attach function

```
> dat$Value
[1] 3 6 4 1 6 2
> dat[,'Value']
[1] 3 6 4 1 6 2
```

```
> attach(dat)
> Value
[1] 3 6 4 1 6 2
```

Vectors can be indexed similarly as data frames

```
> Value[2]
[1] 6
```

basics

Where to go for help?

- A user-friendly intro to R

basics

Where to go for help?

- A user-friendly intro to R
- Several good introductory texts are available - Zuur et al. 2009. A Beginner's Guide to R. Springer.

R basics

Where to go for help?

- A user-friendly intro to R
- Several good introductory texts are available - Zuur et al. 2009. A Beginner's Guide to R. Springer.
- R cheatsheet

basics

Where to go for help?

- A user-friendly intro to R
- Several good introductory texts are available - Zuur et al. 2009. A Beginner's Guide to R. Springer.
- R cheatsheet
- Google is your friend

R basics

Where to go for help?

- A user-friendly intro to R
- Several good introductory texts are available - Zuur et al. 2009. A Beginner's Guide to R. Springer.
- R cheatsheet
- Google is your friend
- Help files for each function using '?function' - may or may not be helpful

R basics

Where to go for help?

- A user-friendly intro to R
- Several good introductory texts are available - Zuur et al. 2009. A Beginner's Guide to R. Springer.
- R cheatsheet
- Google is your friend
- Help files for each function using '?function' - may or may not be helpful
- An intro to R - very detailed
- Ask us!