

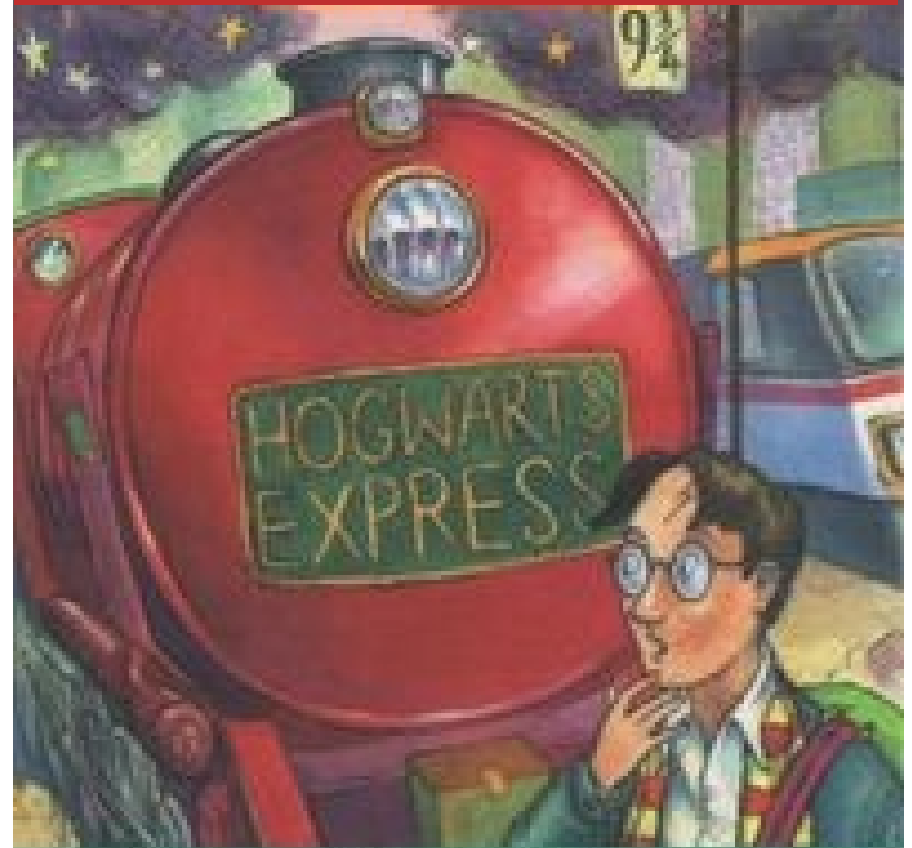
A very brief introduction to R

- Matthew Keller

Some material cribbed from: UCLA Academic Technology Services Technical Report Series (by Patrick Burns) and presentations (found online) by Bioconductor, Wolfgang Huber and Hung Chen, & various Harry Potter websites

R programming language is a lot like magic... except instead of spells you have functions.

R, And the Rise of the Best Software Money Can't Buy



"...this is a terrific book." *The Sunday Telegraph*

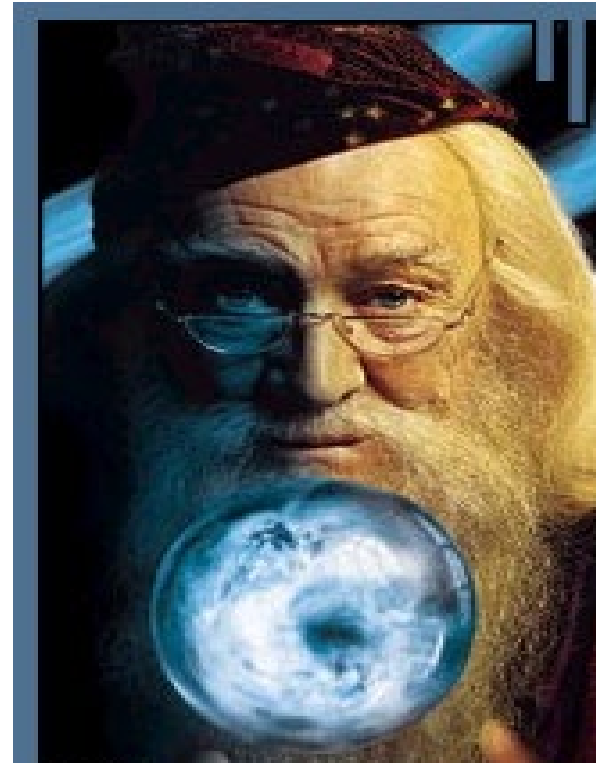


=



muggle

SPSS and SAS users are like muggles. They are limited in their ability to change their environment. They have to rely on algorithms that have been developed for them. The way they approach a problem is constrained by how SAS/SPSS employed programmers thought to approach them. And they have to pay money to use these constraining algorithms.



wizard

R users are like wizards. They can rely on functions (spells) that have been developed for them by statistical researchers, but they can also create their own. They don't have to pay for the use of them, and once experienced enough (like Dumbledore), they are almost unlimited in their ability to change their environment.

History of R

- S: language for data analysis developed at Bell Labs circa 1976
- Licensed by *AT&T/Lucent* to *Insightful Corp.*
Product name: *S-plus*.
- R: initially written & released as an open source software by Ross Ihaka and Robert Gentleman at U Auckland during 90s (R plays on name “S”)
- Since 1997: international R-core team ~15 people & 1000s of code writers and statisticians happy to share their libraries! AWESOME!

“Open source” ... that just means I don't have to pay for it, right?

•No. Much more:

- Provides full access to algorithms and their implementation. Most of R is written in... R, making it easy to see what functions are actually doing.
- Gives the community ability to fix bugs/extend software
- Provides a forum allowing researchers to explore and expand the methods used to analyze data
- Ensures that scientists around the world - and not just ones in rich countries - are the co-owners to the software tools needed to carry out research
- Promotes reproducible research by providing open and accessible tools
- Product of 1000s of leading experts in the fields they know best. It is CUTTING EDGE.**

What is it?

- R is an interpreted computer language.
 - Most user-visible functions are written in R itself, calling upon a smaller set of internal primitives.
 - It is possible to interface procedures written in C, C+, or FORTRAN languages for efficiency, and to write additional primitives.
 - System commands can be called from within R
- R is used for data manipulation, statistics, and graphics. It is made up of:
 - operators (+ - <- * %*% ...) for calculations on arrays & matrices
 - large, coherent, integrated collection of functions
 - facilities for making unlimited types of publication quality graphics
 - user written functions & sets of functions (packages); 16000+ contributed packages so far & growing

R

Advantages

- Fast and free.
- State of the art: Statistical researchers provide their methods as R packages. SPSS and SAS are years behind R!
- 2nd only to MATLAB for graphics.
- Mx, WinBugs, and other programs use R.
- Active user community
- Excellent for simulation, programming, computer intensive analyses, etc.
- Forces you to *think* about your analysis.
- Interfaces with database storage software (SQL)

Disadvantages

R

Advantages

- Fast and free.
- State of the art: Statistical researchers provide their methods as R packages. SPSS and SAS are years behind R!
- 2nd only to MATLAB for graphics.
- Mx, WinBugs, and other programs use R.
- Active user community
- Excellent for simulation, programming, computer intensive analyses, etc.
- Forces you to *think* about your analysis.
- Interfaces with database storage software (SQL)
- Large vectors in 64 bit: 2^{52} length

Disadvantages

- Not user friendly @ start - steep learning curve, minimal GUI.
- No commercial support; figuring out correct methods or how to use a function on your own can be frustrating.
- Working with large datasets is limited by RAM and some operations don't work on vectors $> 2^{31}$ length
- Not natively multi-threaded (easy work-arounds though)
- In the beginning, data prep & cleaning can be messier & more mistake prone in R vs. SPSS or SAS
- Some users complain about hostility on the R listserve

Learning R....



R-help listserve....



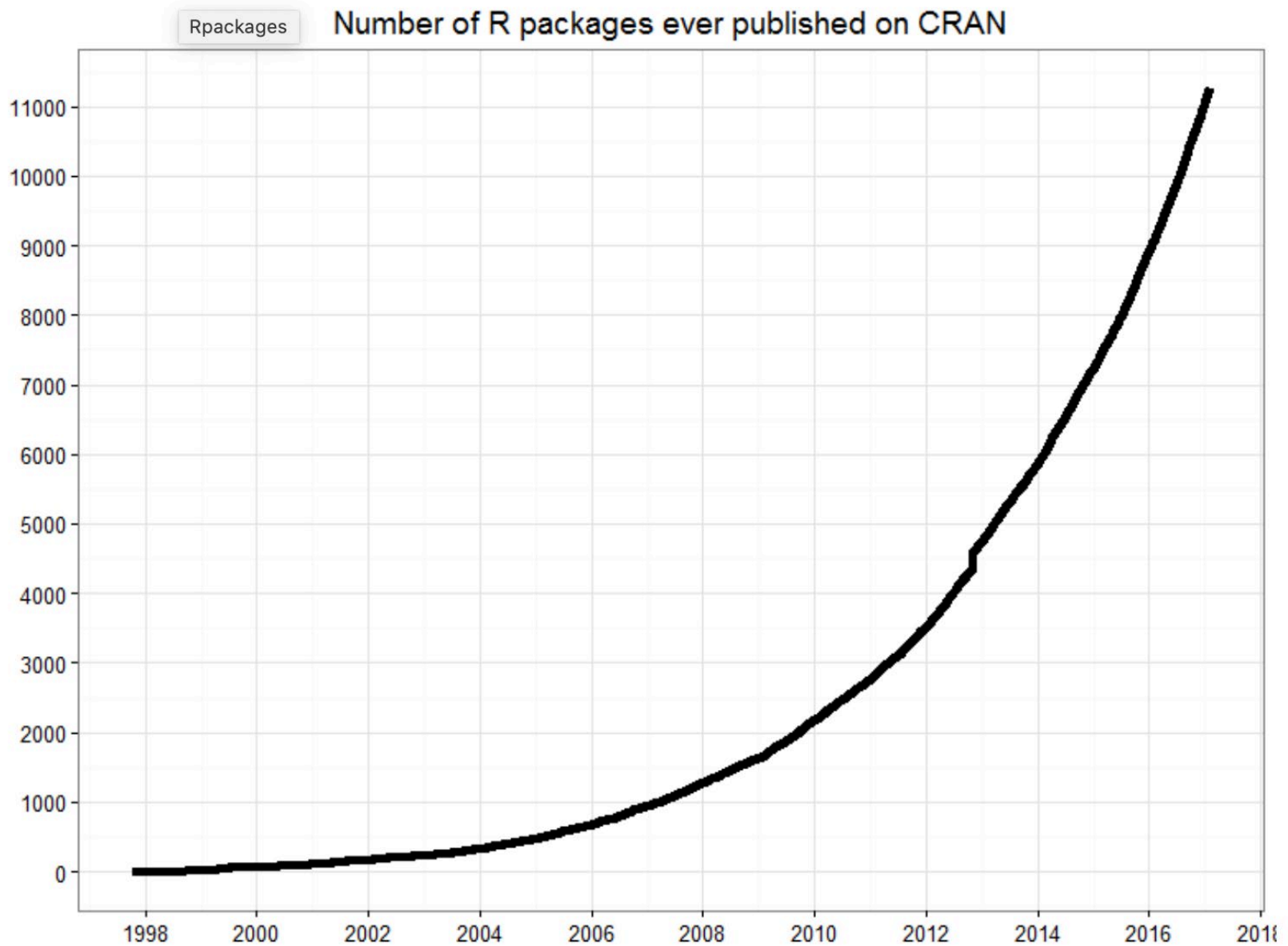
There are over 16K add-on packages

(<http://cran.r-project.org/src/contrib/PACKAGES.html>

<http://www.bioconductor.org> <https://github.com/trending?l=r>)

- This is an enormous advantage - new techniques available without delay, and they can be performed using the R language you already know.
- Allows you to build a customized statistical program suited to your own needs.
- Downside = as the number of packages grows, it is becoming difficult to choose the best package for your needs, & QC is an issue.

Growth of R packages through 2012



Will anything replace R in the future?

- Probably, but it's hard to know when, and I'd be my bottom dollar that it will be an object oriented, open-sourced language like R. (Thus translating your R knowledge will not be tough).
- One possible guess at this next language: JULIA (<http://julialang.org>), which is faster than R, able to work with very large datasets, and has sensible syntax (something R sometimes lacks). It already has 473 packages.



Typical Rstudio session

- Console – output & temporary input - usually unsaved
- Script – tells R what to do. Save this

The screenshot displays the RStudio interface with three main windows:

- Console:** Shows the R version (3.3.3) and a list of help topics such as 'Natural language support but running in an English locale', 'R is a collaborative project with many contributors', and '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.
- Environment:** A table listing functions available in the environment, including 'classx', 'clustered.sample', 'CoreISets', 'dimx', 'geom.mean', 'groups', 'info', 'insert', 'keep', 'look', 'LS', 'lsize', and 'make.Facets', each with its corresponding function signature.
- Script:** A file named 'lecture1.script.2018.R' containing R code for a simulation. The code includes comments and function calls like 'rbinom', 'rnorm', 'sample', and 'mvrnorm'. It describes a simulation where a vector of 5 rows is created, and a matrix of 5 rows by 2 columns is generated. The script ends with 'THE END'.

Environment

- Misc. windows, including help, files, etc.

Typical R session

- R sessions are *interactive*

The screenshot displays the RStudio environment. The top-left pane shows the R version (3.3.3) and copyright information. The top-right pane shows a script editor with R code for a lecture. The bottom-left pane shows the Environment pane with a list of functions. The bottom-right pane shows the R Documentation for the `mvrnorm` function.

```
lecture1.script.2018.R
563 # The third column is binary, with overall prob(x=1) = .25
564 # Give each column an informative name
565
566 # (b) Plot a histogram of the first two variables. The do a
567 # scatterplot of the first two variables against each other.
568 # See ?hist and ?plot
569
570 # (c) What is the mean, median, & var of the first two
571 # columns? What % of the third column is equal to 1? See ?mean
572 # ?median ?var ?summary
573
574 # (d) What is the mean, median, & variance of the first two
575 # columns WHEN the third column is equal to 1? Try
576 # creating a vector that is TRUE when the third column is 1 and
577 # FALSE when 0, then use this vector as an index). Place the
578 # mean, median, & var for each of these column into two 3
579 # element vectors named hw4d1 and hw4d2 respectively
580
581 # (e) Take a random sample of 50 rows of hw4a (without
582 # replacement) using the "sample" function. What is the mean &
583 # standard deviation (see ?sd) of the second column for this
584 # subset of rows from hw4a? Place this information into a 2
585 # element vector named hw4e
586
587 # (f) We'd like to do a monte carlo experiment where we
588 # randomly sample 50 rows (without replacement) like above, but
589 # do this 5 times. Each time, find the mean and sd of the first
590 # column of hw4a and place this information in a row of a
591 # matrix of 5 rows by 2 columns named "hw4f". In the end, the
592 # first column of hw4f should contain 5 resampled means and the
593 # second column 5 resampled sd's. This is a (small) sampling
594 # distribution of means and sd's from the original hw4a
595 # population data. Eventually, we'll do this much more
596 # efficiently using a for loop, but for now...
```

Write small bits of code here and run it

Typical R session

- R sessions are *interactive*

The screenshot shows the RStudio environment. The top pane is the script editor, displaying a file named 'lecture1.script.2018.R'. The code in the script includes comments and R commands for data simulation and analysis. The bottom-left pane is the console, showing the R version (3.3.3) and various help messages. The bottom-right pane is the environment, showing a list of functions available in the current session.

```
column); see ?rbinom to help in choosing the random 10% of
scores
563 # The third column is binary, with overall prob(x=1) = .25
564 # Give each column an informative name
565
566 # (b) Plot a histogram of the first two variables. The do a
scatterplot of the first two variables against each other.
See ?hist and?plot
567
568 # (c) What is the mean, median, & var of the first two
columns? What % of the third column is equal to 1? See ?mean
?median ?var ?summary
569
570 # (d) What is the mean, median, & variance of the first
column is equal to 1? See ?mean ?median ?var ?summary
571
572 # (e) Take a random sample of hw4a (without
replacement) using the sample() function. What is the mean &
standard deviation of the first column for this
subset of rows from hw4a? Place this information into a 2
element vector named hw4e
573
574 # (f) We'd like to do a monte carlo experiment where we
randomly sample 50 rows (without replacement) like above, but
do this 5 times. Each time, find the mean and sd of the first
column of hw4a and place this information in a row of a
matrix of 5 rows by 2 columns named 'hw4f'. In the end, the
first column of hw4f should contain 5 resampled means and the
second column 5 resampled sd's. This is a (small) sampling
distribution of means and sd's from the original hw4a
population data. Eventually, we'll do this much more
efficiently using a for loop, but for now...
575
```

Write small bits of code here and run it

Output appears here. Did you get what you wanted?

Typical R session

- R sessions are *interactive*

The screenshot shows the RStudio interface with the following components:

- Console:** Displays the R version (3.3.3) and a list of functions in the environment.
- Script Editor:** Contains R code for a simulation experiment, including comments and function calls like `sample` and `mvrnorm`.
- Environment Pane:** Lists functions such as `classx`, `clustered.select`, `CoreISets`, `dimx`, `geom.mean`, `groups`, `info`, `insert`, `keep`, `look`, `LS`, `lsize`, and `make.Facets`.

Output appears here.
Did you get what you wanted?

Adjust your syntax here depending on this answer.

Typical R session

- R sessions are *interactive*

The screenshot displays the RStudio environment. The top pane shows a script file named 'lecture1.script.2018.R'. The script contains several lines of R code, including comments and function calls. The console pane on the left shows the R version (3.3.3) and copyright information. The environment pane at the bottom left lists various functions and their arguments. The bottom right pane shows the R Documentation for the 'mvnrm' function, which is used to simulate from a multivariate normal distribution.

```
566 # (c) What is the mean, median, & variance of the first two
567 # columns WHEN the third column is equal to 1? See ?mean
568 # (d) What is the mean, median, & variance of the first two
569 # columns WHEN the third column is equal to 1? See ?mean
570 # (e) Take a random sample of 50 rows of hw4a (without
571 # replacement) using the "sample" function. What is the mean &
572 # standard deviation (see ?sd) of the second column for this
573 # subset of rows from hw4a? Place this information into a 2
574 # element vector named hw4e
```

The environment pane shows the following functions:

Function	Description
classx	function (dd)
clustered.select	function (x, num = 1)
CoreISets	function (x1 = rnorm(100, 15, 5), ndistr = 3, coefs = c(0, ...))
dimx	function (dd)
geom.mean	function (x)
groups	function (x)
info	function (x)
insert	function (x, z, pos, col = TRUE)
keep	function (... , list = character(0), sure = FALSE)
look	function (x, type = 1, number = 50)
LS	function (space = 1, pattern = "")
lsize	function (space = 1, pattern = "", sort = TRUE)
make.Facets	function (x)

The R Documentation pane shows the following information for the 'mvnrm' function:

Simulate from a Multivariate Normal Distribution

Description
Produces one or more samples from the specified multivariate normal distribution.

Usage
mvnrm(n = 1, mu, Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)

Arguments

Typical R session

- R sessions are *interactive*

The screenshot displays the RStudio environment. The top-left pane shows the R version (3.3.3) and copyright information. The top-right pane shows a script editor with R code for a simulation. The bottom-left pane shows the Environment window with a list of functions. The bottom-right pane shows the Help window for the `mvnorm` function.

```
lecture1.script.2018.R
column); see ?rbinom to help in choosing the random 10% of
scores
563 # The third column is binary, with overall prob(x=1) = .25
564 # Give each column an informative name
565
566 # (b) Plot a histogram of the first two variables. The do a
scatterplot of the first two variables against each other.
See ?hist and?plot
567
568 # (c) What is the mean, median, & var of the first two
columns? What % of the third column is equal to 1? See ?mean
?median ?var ?summary
569
570 # (d) What is the mean, median, & variance of the first two
columns WHEN the third column is equal to 1? (hint: first try
creating a vector that is TRUE when the third column is 1 and
FALSE when 0, then use this vector as an index). Place the
mean, median, & var for each of these column into two 3
element vectors named hw4d1 and hw4d2 respectively
571
572 # (e) Take a random sample of 50 rows of hw4a (without
replacement) using the "sample" function. Find the mean &
standard deviation (see ?sd) of the second column for this
subset of rows from hw4a? Place this information into a 2
element vector named hw4e
573
574 # (f) We'd like to do a monte carlo experiment where we
randomly sample 50 rows (without replacement) like above, but
do this 5 times. Each time, find the mean and sd of the first
column of hw4a and place this information in a row of a
matrix of 5 rows by 2 columns named "hw4f". In the end, the
first column of hw4f should contain 5 resampled means and the
second column 5 resampled sd's. This is a (small) sampling
distribution of means and sd's from the original hw4a
population data. Eventually, we'll do this much more
efficiently using a for loop, but for now...
575
626.1 THE END
```

Environment

Function	Class
classx	function (dd)
clustered.select	function (x, num = 1)
CoreISets	function (x1 = rnorm(100, 15, 5), ndistr = 3, coefs = c(0...
dimx	function (dd)
geom.mean	function (x)
groups	function (x)
info	function (x)
insert	function (x, z, pos, col = TRUE)
keep	function (... , list = character(0), sure = FALSE)
look	function (x, type = 1, number = 50)
LS	function (space = 1, pattern = "")
lsize	function (space = 1, pattern = "", sort = TRUE)
make.facets	function (x)

Help

R: Simulate from a Multivariate Normal Distribution

`mvnorm` (MASS)

Simulate from a Multivariate Normal Distribution

Description

Produces one or more samples from the specified multivariate normal distribution.

Usage

```
mvnorm(n = 1, mu, Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
```

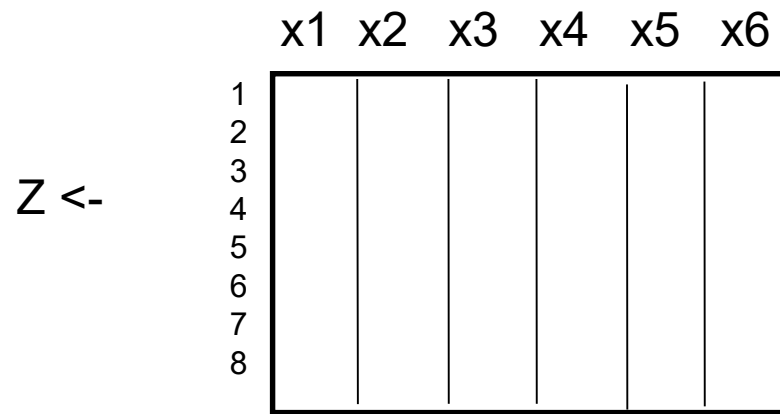
Arguments

At end, all you need to do is save your script file(s) - which can easily be rerun later.

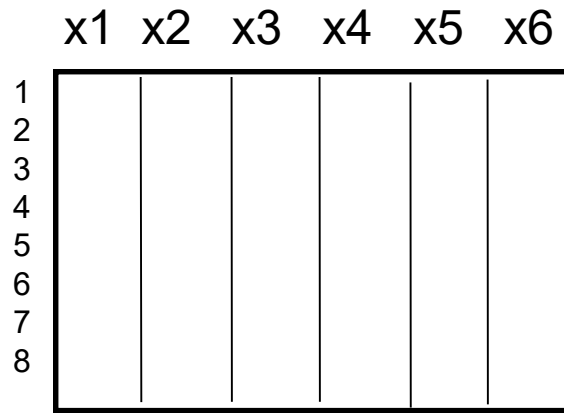
R Objects

- Almost all things in R – functions, datasets, results, etc. – are OBJECTS.
 - (graphics are written out and are not stored as objects)
- Script can be thought of as a way to make objects. Your goal is usually to write a script that, by its end, has created the objects (e.g., statistical results) and graphics you need.
- Objects are classified by two criteria:
 - MODE: how objects are stored in R - character, numeric, logical, list, & function
 - CLASS: how objects are treated by functions (important to know!) - [vector], matrix, array, factor, data.frame, & 1000s of special classes created by specific functions

R Objects



R Objects



The MODE of Z is determined automatically by the types of things stored in Z – numbers, characters, etc. Vectors & matrices must have their values all of the same mode. Lists can be a mix of modes.

R modes (to check, use mode() function):

numeric – numbers

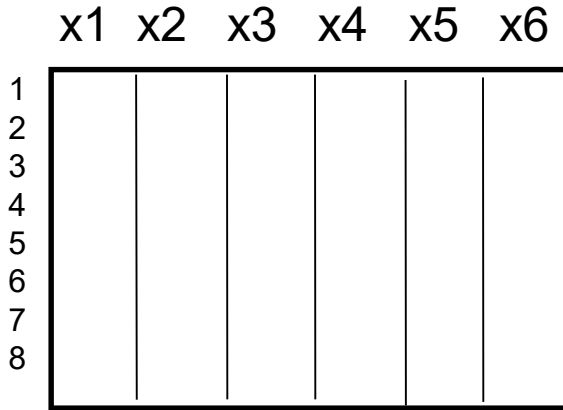
character

list – a concatenation of elements of different modes

logical – TRUE/FALSE

function

R Classes



The CLASS of Z is either set by default depending, on how it was created, or is explicitly set by user. You can check the objects' class and change it. It determines how functions deal with Z. If of class "lm", R searches for a function fun.lm

NOTE: If an object has two classes - c("first", "second") - R searches for a function called fun.first and, if it finds it, applies it to the object. If no such function is found, a function called fun.second is tried. If no class name produces a suitable function, the function fun.default is used.

R classes (to check, use class() function):

[for **vectors**, mode & class are same] - **logical**, numeric, **character**

[modes & class are same for these 2 as well] - **function**, **list** (when generic)

factor

matrix

array

data.frame

Learning R

- Read through the CRAN website & intro manual
- Know your objects' modes & classes: `mode(x)`; `class(x)`
- Because R is interactive, errors are your friends!
- `?lm` gives you help on `lm` function. Reading help files can be very... helpful
- **MOST IMPORTANT** - the more time you spend using R, the more comfortable you become with it. After doing your first real project in R, you won't look back. I promise.

Recommended Book

- An R and S-PLUS Companion to Applied Regression: An excellent overview of R, not just regression in R. Highly recommended. Many of the HWs we will do were inspired by Fox's book. If you are the type of person who likes to have a book, buy this one. \$56 at Amazon.

