Sharing the Love

Building Your First Package

Chris Campbell

MANGOSOLUTIONS
data analysis that delivers

MANGOSOLUTIONS
data analysis that delivers

Agenda

- What is a Package?
- How do I Build a Package?
- Unit Testing
- Sharing a Package

What is a Package?

MANGOSOLUTIONS
data analysis that delivers

- R is like a giant combining robot


```
> search()
[1] ".GlobalEnv"        "package:stats"
[3] "package:graphics"  "package:grDevices"
[5] "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"
[9] "package:base"
```

What is a Package?

MANGOSOLUTIONS
data analysis that delivers

- R is like a giant combining robot
- Add units of functionality

```
> install.packages("MSToolkit", lib = .libPaths()[1],
+     repos = "http://cran.ma.imperial.ac.uk/")

> require(MSToolkit)
```

What is a Package?

MANGOSOLUTIONS
data analysis that delivers

- Add units of functionality
- Until ultimate functionality is achieved

```
> search()
 [1] ".GlobalEnv"        "package:MSToolkit"
 [3] "package:MASS"      "package:stats"
 [5] "package:graphics"  "package:grDevices"
 [7] "package:utils"     "package:datasets"
 [9] "package:methods"   "Autoloads"
[11] "package:base"
```

What is a Package? Convenience

MANGOSOLUTIONS
data analysis that delivers

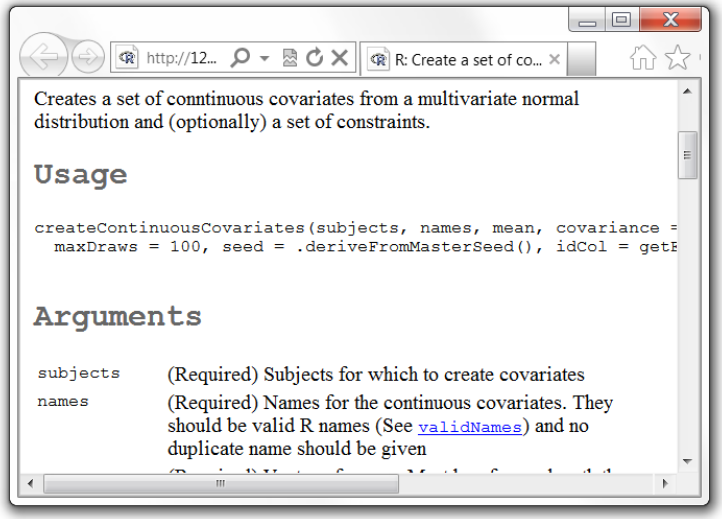
- Multiple units of code can be accessed together
- Interdependent units of code can be shared together
- Interdependent units of code can be tested together

What is a Package? Communication

MANGOSOLUTIONS
data analysis that delivers

- Scripts may not be clear to others
- We can define workflow or process with reusable content
- High level functions to combine analysis steps
- Help files
- Demonstration function

```
> help("createContinuousCovariates")
```



What is a Package? Reproducible Results

MANGOSOLUTIONS
data analysis that delivers

- Scripts can be messy and hard to reuse
- Functions should use variables, rather than hard coded values
- Since fewer manual steps, the workflow is standardised

What is a Package? Memory Efficient Functionality

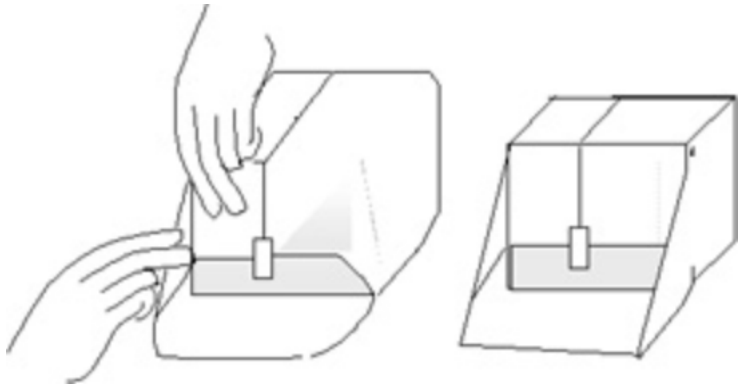
MANGOSOLUTIONS
data analysis that delivers

- Systems can become bloated with excess functionality
- Currently 4301 packages on CRAN
- Choose only essential tools
- Rest is not loaded

How do I make a Package?

MANGOSOLUTIONS
data analysis that delivers

- Rtools (and R)
- Set PATH
- Create files with `package.skeleton`
- Compile package



How do I make a Package? Rtools

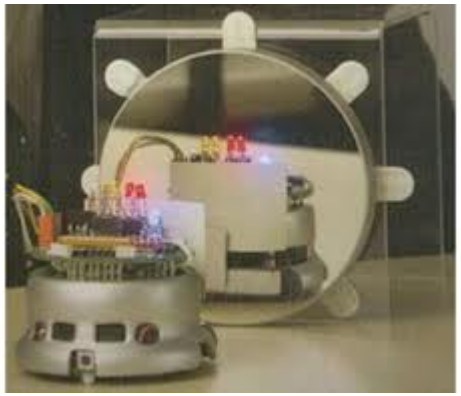
MANGOSOLUTIONS
data analysis that delivers

- Building R for Windows
- http://cran.ma.imperial.ac.uk/bin/windows/Rtools/
- Collated by Prof Brian Ripley
- Maintained by Duncan Murdoch

How do I make a Package? Set PATH

MANGOSOLUTIONS
data analysis that delivers

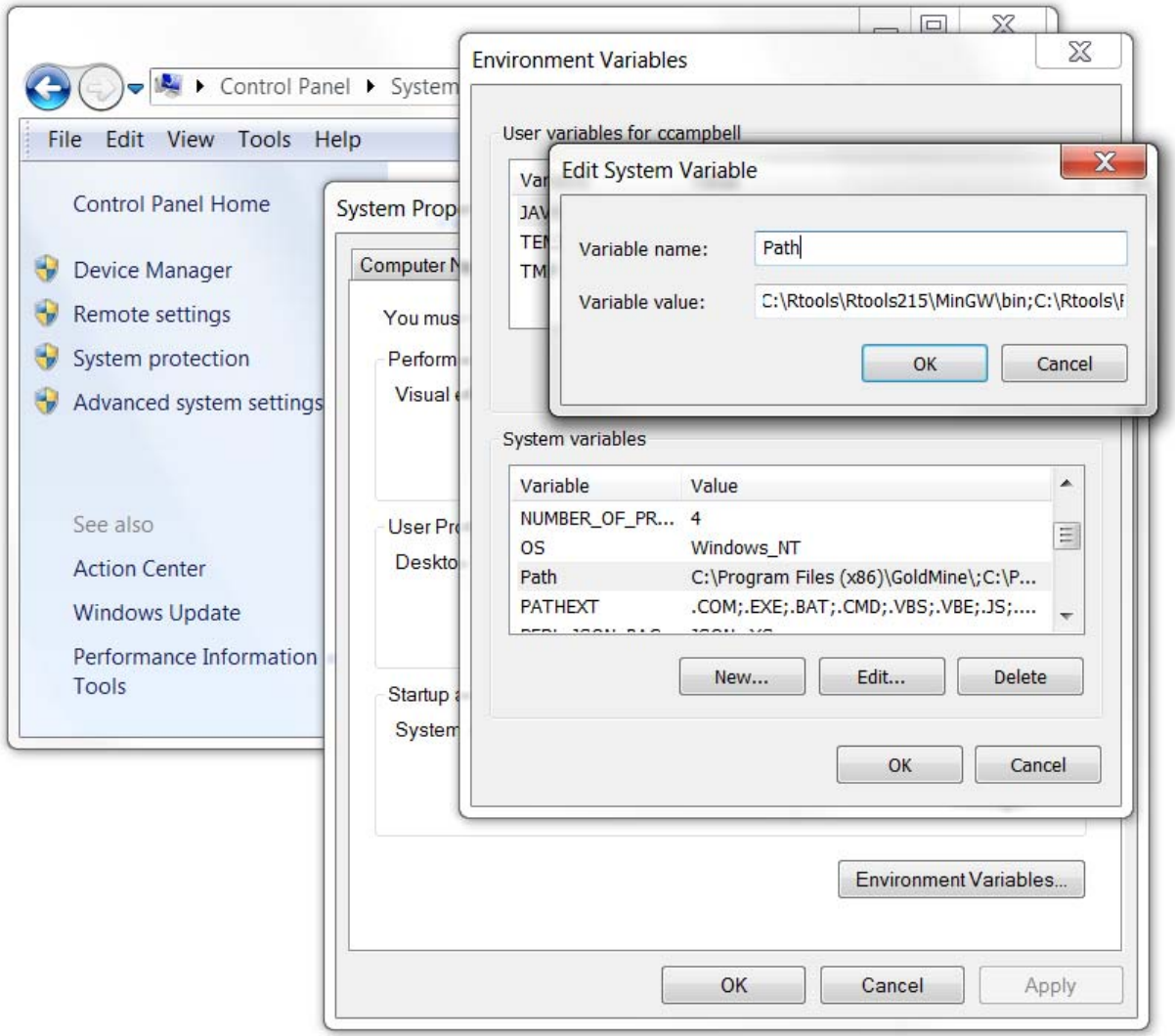
- Computers are not self-aware (yet)
- They need to be told where to find programs
- The list of places looks is the PATH



How do I make a Package? Set PATH

MANGOSOLUTIONS
data analysis that delivers

- Add **bin** folders of Rtools and R, separated by ; e.g.
- ;C:\Rtools\Rtools215\MinGW\bin;C:\Rtools\Rtools215\bin;C:\Program Files\R\R-2.15.1\bin\i386



How do I make a Package? package.skeleton

MANGOSOLUTIONS
data analysis that delivers

- A package consists of files & folders following a strict structure
- DESCRIPTION describes the package!
- NAMESPACE defines visible functions in the package
- Rd files are help files in LaTaX markup; use tags with syntax `\tag{text}` to add text and formatting to help documents
- These files can be created for you by `package.skeleton`

How do I make a Package? package.skeleton

MANGOSOLUTIONS
data analysis that delivers

```
> # a function to calculate elapsed time
> # the Format must be the same for Start and Time
> # see ?strptime for format codes

> getElapsedTime <- function(Start, Time,
+     Format = "%m/%d/%y %H:%M") {
+
+     startT <- as.POSIXct(Start, format = Format)
+     timeT <- as.POSIXct(Time, format = Format)
+     elapsedT <- timeT – startT
+
+     cat("Time difference in", attr(elapsedT, "units"), "\n")
+     return(c(elapsedT))

> ls(".GlobalEnv")
[1] "getElapsedTime"
```

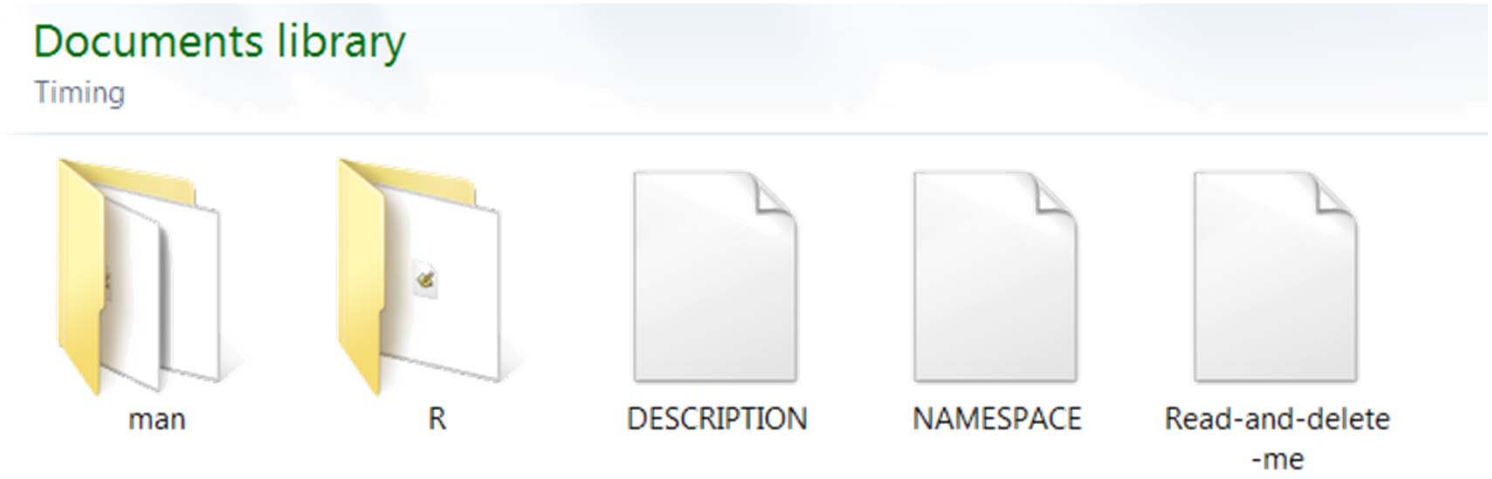
How do I make a Package? package.skeleton

MANGOSOLUTIONS
data analysis that delivers

```
> package.skeleton("Timing")
Creating directories ...
Creating DESCRIPTION ...
Creating NAMESPACE ...
Creating Read-and-delete-me ...
Saving functions and data ...
Making help files ...
Done.
Further steps are described in './Timing/Read-and-delete-me'.
```

How do I make a Package? package.skeleton

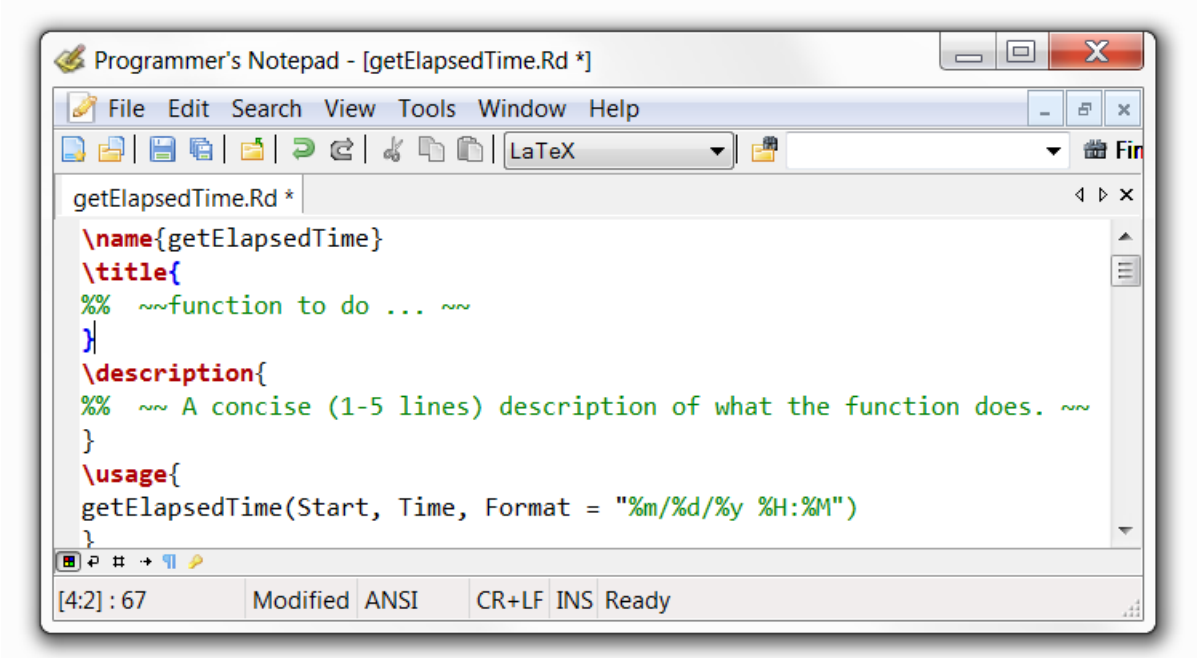
MANGOSOLUTIONS
data analysis that delivers



How do I make a Package? package.skeleton

MANGOSOLUTIONS
data analysis that delivers

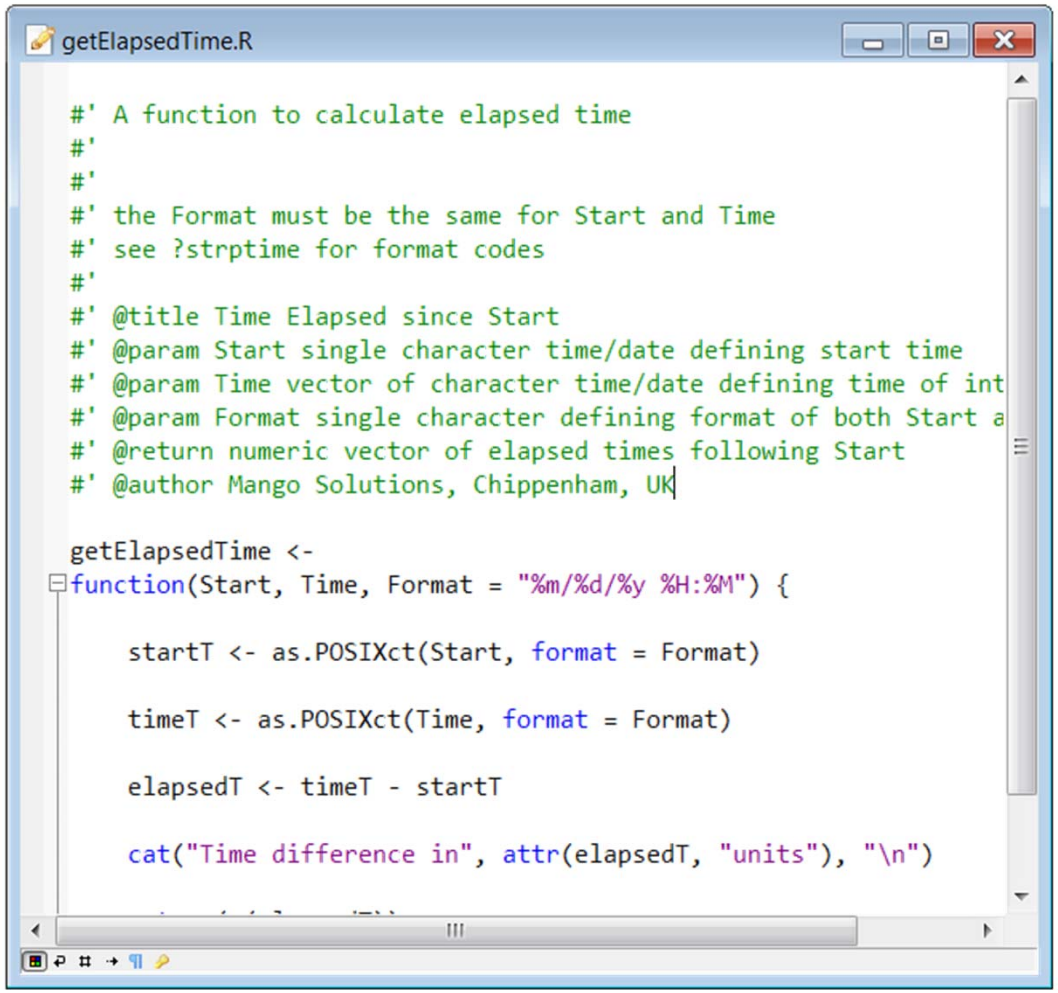
- Edit help files



How do I make a Package? package.skeleton

MANGOSOLUTIONS
data analysis that delivers

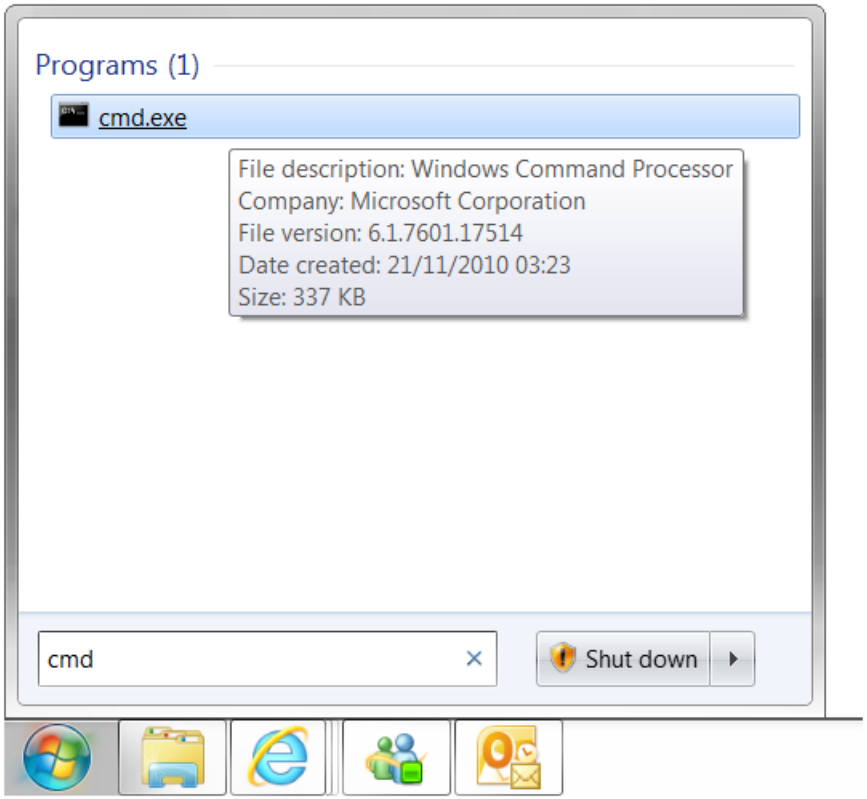
- Or use roxygen2 turn flags inline in code into Rd files



How do I make a Package? Compile

MANGOSOLUTIONS
data analysis that delivers

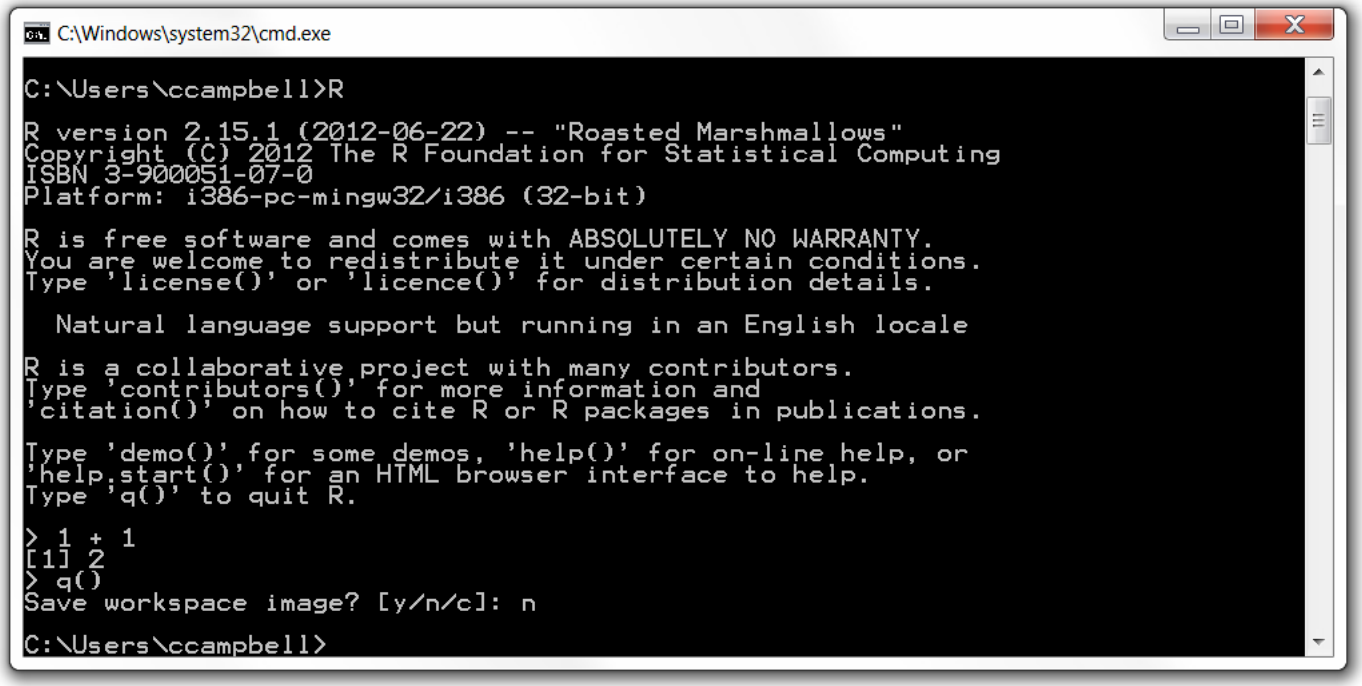
- R on PATH is available from **cmd**



How do I make a Package? Compile

MANGOSOLUTIONS
data analysis that delivers

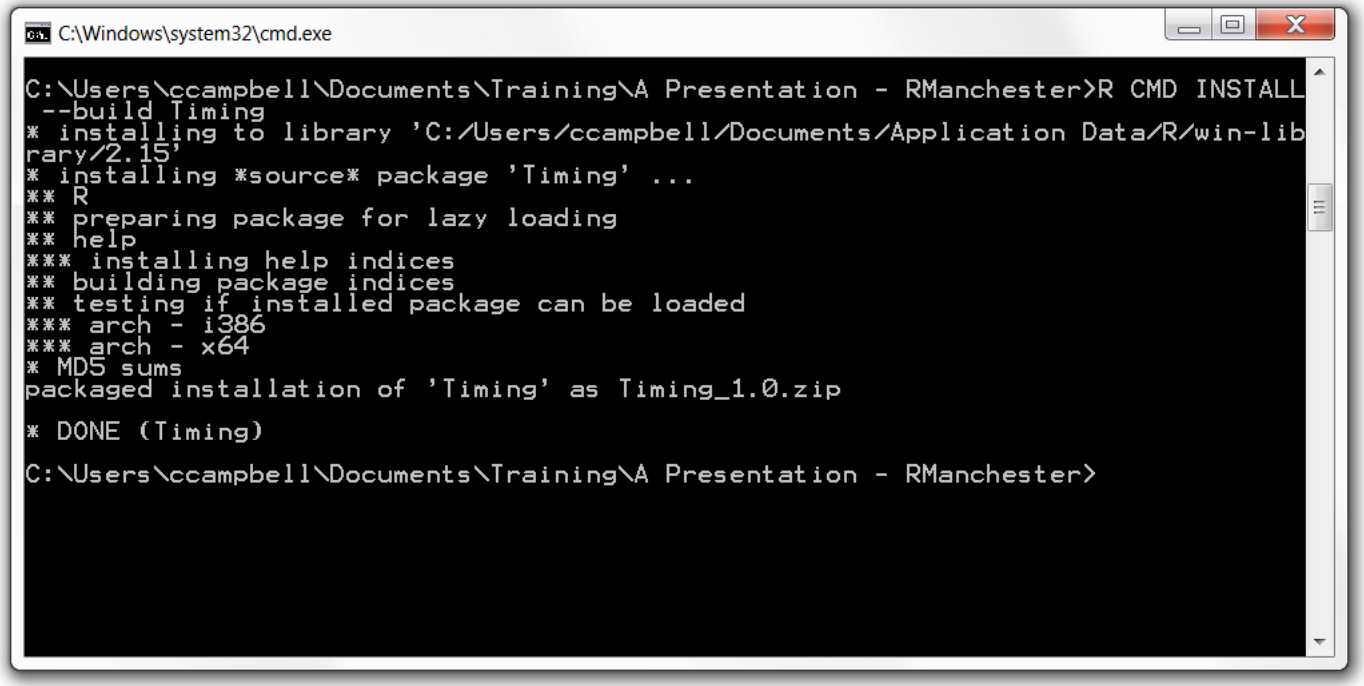
- R on PATH is available from **cmd**



How do I make a Package? Compile

MANGOSOLUTIONS
data analysis that delivers

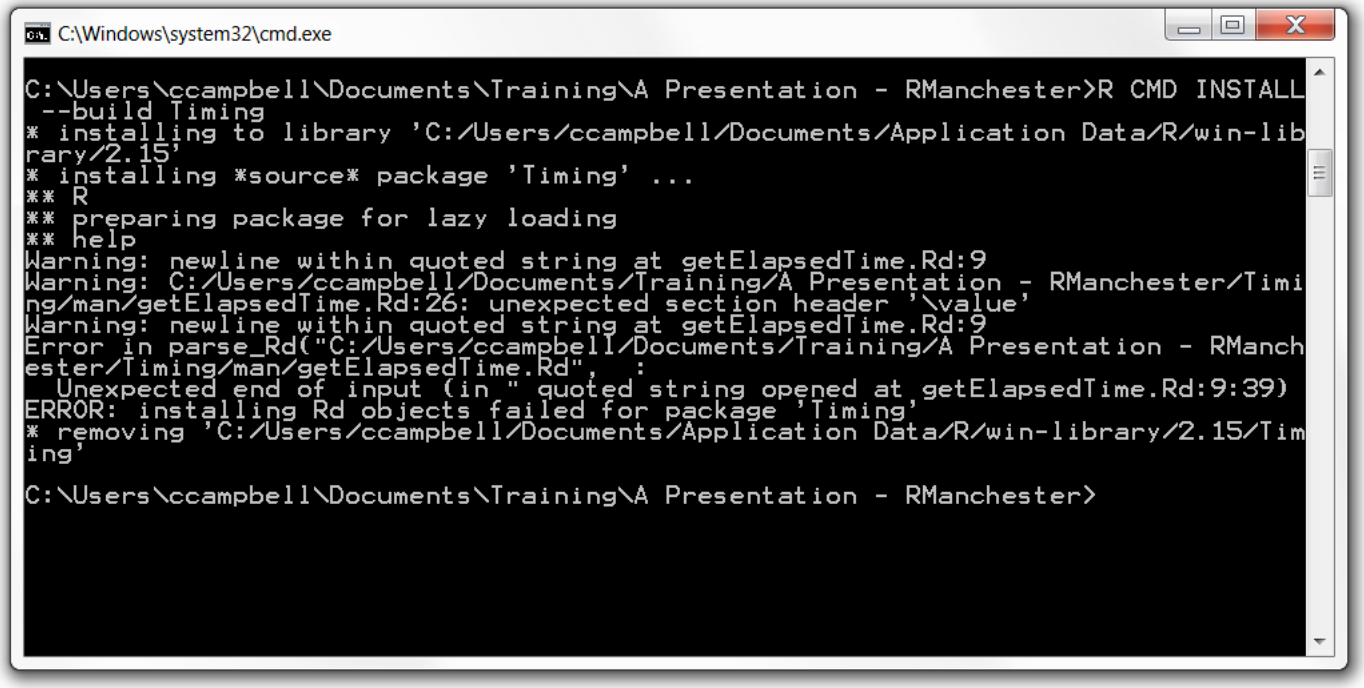
- R CMD INSTALL --build Timing



How do I make a Package? Might take a few tries!

MANGOSOLUTIONS
data analysis that delivers

- Rd files can be tricky to get right manually!



How do I make a Package? Versions

MANGOSOLUTIONS
data analysis that delivers

- Our code is important; we must keep track of it!
- Manually – change filename with every save
- Automatically - use SVN, GIT

Unit Testing
Check it Works

MANGOSOLUTIONS
data analysis that delivers

- Unit Testing is an important part of coding
- Try packages testthat or RUnit
- Check code does what we expect

```
> getElapsedTime(Start = "05/21/12 15:20",
+     Time = c("05/21/12 15:20", "05/21/12 15:21",
+         "05/21/12 15:30", "05/21/12 16:20"))
Time difference in secs
[1]    0   60  600 3600
```

Unit Testing Checks

MANGOSOLUTIONS
data analysis that delivers

- Test for each function
- Contains multiple checks

```
test.getElapsedTime <- function()
{
test1 <- getElapsedTime(Start = "05/21/12 15:20",
    Time = c("05/21/12 15:20", "05/21/12 15:21",
        "05/21/12 15:30", "05/21/12 16:20"))
checkEquals(test1, c(0, 60, 600, 3600),
    msg = " TEST 1: Default example)
...
```

Unit Testing
Automated Tests

MANGOSOLUTIONS
data analysis that delivers

- Run all tests at once

```
runTests <- function(testPath =
    system.file(package = "Timing", "testing")
{
    testSuite <- defineTestSuite(name = "Unit Test Suite",
        dirs = testPath,
        testFileRegexp = "^runit.+\\.[rR]$",
        testFuncRegexp = "^test.+" )
}
```

Unit Testing Robustness

MANGOSOLUTIONS
data analysis that delivers

- Unit Testing is an important part of coding
- Function may be passed data outside it's anticipated structure
- We don't want to create misleading results

```
> getElapsedTime(c("05/21/12 12:01", "05/21/12 18:03"),
+     "05/21/12 15:20")
Time difference in hours
[1]  3.316667 -2.716667
```

Unit Testing
Errors are Good!

MANGOSOLUTIONS
data analysis that delivers

- We expects a function to behave in a certain way
- We can use `stop` and `warning` to flag problems

```
getElapsedTime <- function(Start, Time,
    Format = "%m/%d/%y %H:%M") {

    if (any(length(Start) != 1))) {
        stop("Start should be length 1") }
...
```

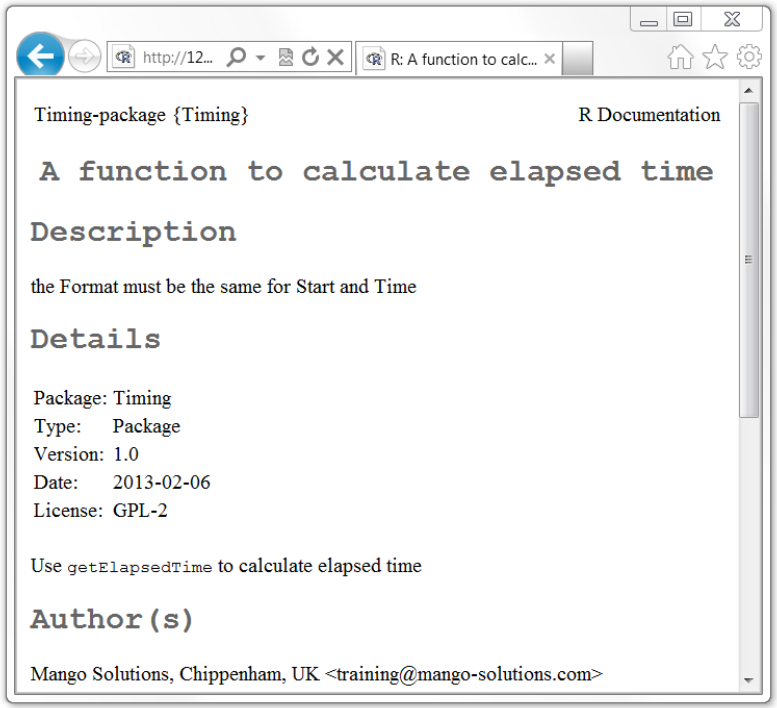
Sharing a Package
It Works!

MANGOSOLUTIONS
data analysis that delivers

- Now you can share your code

```
> require(Timing)

> help(package = "Timing")
```



Sharing a Package Giving Back to the Community

MANGOSOLUTIONS
data analysis that delivers

- With colleagues & peers as a zip
- Globally on r-forge, CRAN

Timing_1.0.zip

MANGOSOLUTIONS
data analysis that delivers

Summary

- Packages allow us to collate code
- We are encouraged to make scripts work generically, so analysis more reproducible
- We can share ideas with others, allowing them to use our methods
- We are encouraged to split our scripts into units that can be efficiently tested