

# Using RMarkdown for reproducible and neat documents

Your name here  
Your affiliation here

## Contents

<b>Overview</b>	<b>3</b>
Use different headings . . . . .	3
Like this subheading . . . . .	3
<b>Just like <math>\text{\LaTeX}</math>, but <i>more versatile</i>.</b>	<b>3</b>
Define equations . . . . .	4
Embed images/gifs: . . . . .	4
Create, alter, and embed plots . . . . .	5
Show plots with associated code . . . . .	6
And tables . . . . .	8
Embed code from different languages . . . . .	9
This is R code . . . . .	9
<b>shell/bash</b> . . . . .	9
Octave (and MATLAB from the RMatlab package). . . . .	9
HTML . . . . .	9
CSS . . . . .	10
Javascript to access <b>html</b> and <b>css</b> . . . . .	10
Python . . . . .	10
Here's a complete list of available languages . . . . .	10

Date: 2019-10-20  
R version: 3.5.0  
\*Corresponding author: your email here

## Overview

This document showcases how to create and use **RMarkdown** documents.

You can write in-line `code` if you want to differentiate between when you are typing normally or highlighting `model parameters`, for example.

Equations like this  $t' = \gamma(t - vx/c^2)$ , to appear within text lines.

Create links to your website.

Make footnotes<sup>1</sup>.

## Use different headings

Like this subheading

**Just like  $\text{\LaTeX}$ , but *more versatile*.**

---

<sup>1</sup>Where the footnote goes here and it is automatically formatted

## Define equations

Accordingly, we write the eigenfunction of a spinless particle as the superposition of plane wave states of momentum ( $\pi$ ) and energy ( $E_j$ ) having amplitudes  $a(\pi, E_j)$

$$\phi(r, t) = \sum_{i,j} a(p_i, E_j) e^{\frac{i}{\hbar}(p_i \cdot r - E_j t)}$$

## Embed images/gifs:



Create, alter, and embed plots

### Some random data

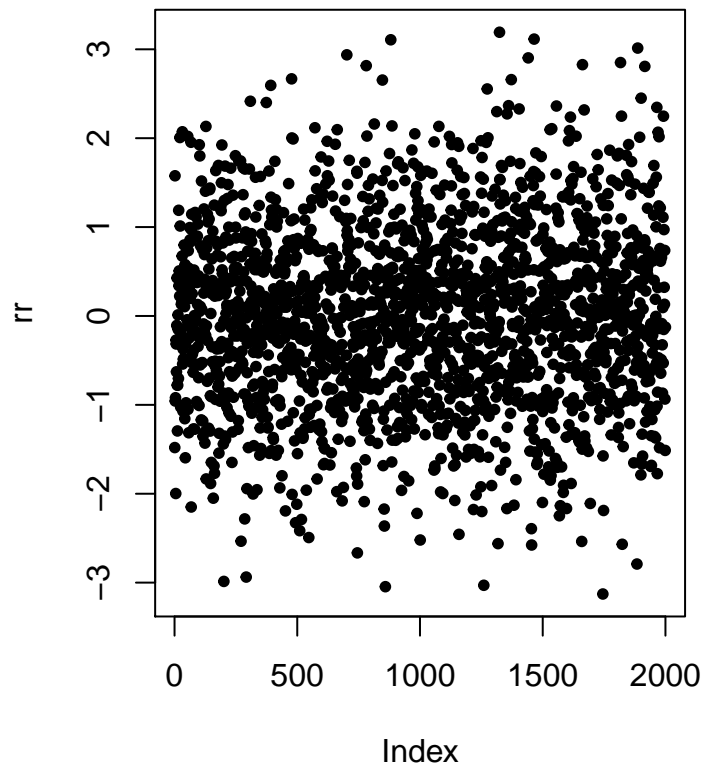


Figure 1. Example of a stock plot embedded into a PDF from RMarkdown.

## Show plots with associated code

```
require(viridis)
bm <- 0
par(las = 1, bty = "n")
xlim <- c(-5, 5)
ylim <- c(0, 0.5)
set.seed(12)
N <- 2000
rr <- rnorm(N)
rr2 <- rnorm(N^2)
rr3 <- rnorm(N + 0.3)
rrd <- density(rr)
rrd2 <- density(rr2)
rrd3 <- density(rr3)
main <- paste0(N, " points but plot better")
xlab <- "Points in space"
if (bm == 1) {
  layout(matrix(c(rep(1, 3), 2:4), 2, 3, byrow = TRUE))
  sc <- 1
  plot(rr, las = 1, bty = "n", col = adjustcolor(viridis(N), 0.5), pch = 20, cex = runif(1))
  main = main, xlab = xlab)
  for (r in list(rrd, rrd2, rrd3)) {
    plot(r, xlim = xlim, ylim = ylim, main = "")
    polygon(r, col = adjustcolor(viridis(250)[sc], 0.5), border = viridis(250)[sc])
    sc <- sc + 100
  }
} else {
  par(mfrow = c(1, 1))
  plot(rr, las = 1, bty = "n", col = adjustcolor(viridis(N), 0.5), pch = 20, cex = runif(1))
  main = main, xlab = xlab)
}
```

## 2000 points but plot better

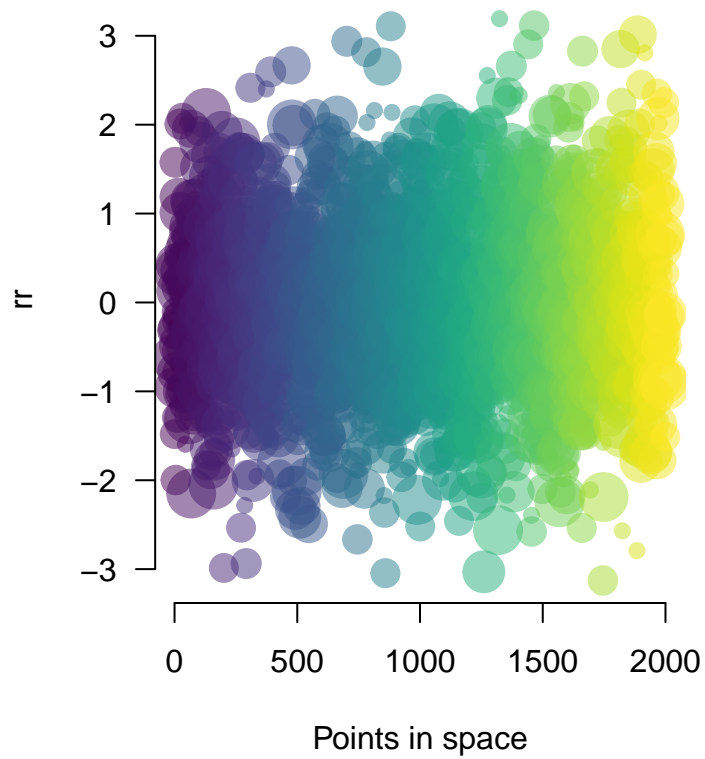


Figure 2. Example of a plot with improved graphics and its associated code embedded into a PDF from RMarkdown.

## And tables

Table 1. Definitions of model parameters for individual hosts and **parasites**.  
Dimensions and units: -, dimensionless; cm, centimetres; J, Joules; L, length.

Parameter	Definition	Dimension(unit)
$L$	structural length	cm
$ee$	scaled reserve density	J (cm <sup>3</sup> )
$D$	host development	—
$RH$	energy in reproduction buffer	J



## Embed code from different languages

This is R code

```
if (pck == 1) {  
  p <- c("rJava", "RNetLogo")  
  remove.packages(p)  
  # then install rJava and RNetLogo from source  
  install.packages("rJava", repos = "https://cran.r-project.org/")  
  install.packages("RNetLogo", repos = "https://cran.r-project.org/")  
}
```

shell/bash

```
echo "Hello Bash!"  
pwd # check working dir  
git init # initialise git
```

Octave (and MATLAB from the RMatlab package).

RMatlab documentation.

```
b = [4; 9; 2] # Column vector  
A = [ 3 4 5;  
      1 3 1;  
      3 5 9 ]  
x = A \ b     # Solve the system Ax = b
```

HTML

```
<!-- links-->  
  <div class="footer">  
    <a href="dd_feed.html"  
      class="transition fade_in">  
      Latest post  
    </a>  
    &nbsp; &nbsp; &nbsp; &nbsp;  
    <a href="dd_contact.html"  
      class="transition fade_in">  
      Contact  
    </a>  
    &nbsp; &nbsp; &nbsp; &nbsp;
```

```

        <a href="dd_subscribe.html"
        class="transition fade_in">
            Subscribe
        </a>
    </div>

```

## CSS

```

body {
    color: red;
}

```

## Javascript to access html and css

```

$('.title').css('color', 'red')

```

## Python

```

x = 'hello, python world!'
print(x.split(' '))

```

## Here's a complete list of available languages

```

names(knitr::knit_engines$get())

```

```

## [1] "awk"      "bash"     "coffee"  "gawk"     "groovy"   "haskell"  "lein"
## [9] "node"     "octave"   "perl"     "psql"     "Rscript"  "ruby"     "sas"
## [17] "sed"      "sh"       "stata"    "zsh"      "highlight" "Rcpp"     "tikz"
## [25] "c"        "fortran"  "fortran95" "asy"      "cat"      "asis"     "stan"
## [33] "block2"   "js"       "css"      "sql"      "go"       "python"   "julia"
## [41] "scss"

```