Introductory guide to using R Markdown

Emma McLarnon

13 October 2017

This is an introductory guide to using R Markdown for beginners.

To begin, in RStudio go to file > new file > R Markdown. This will give the output format option, select the output you want (for this document, I've chosen html). You will be presented with a new page with the title, author etc.

To begin using R Markdown, you need to insert a "chunk". This is a chunk of code. You can use the insert button (located next to the run button in the page window) or you can type three back ticks followed by r and curly braces: e.g. "'r { }. You will then state some parameters inside these braces, for example the data set and if you want the code and output to be displayed.

I will go through what some of the main parameters in the curly braces do. These parameters control what is displayed in the output.

1. Main chunk options

There are a number of chunk options, which control how the code is displayed in the final document. There are 3 main options you may want to control:

1. {r include = TRUE} means that you include the code chunk in the output. If you choose {r include = FALSE}, this will suppress any code and corresponding results within a particular chunk of code, but it will still display any text you type.
2. {r echo = T} displays both the code and the results. {r echo= F} displays only the results.
3. {r eval= T}

1. {r include=FALSE}{r include = TRUE}

This parameter means code and results aren't presented in the finished file. Below, I have specified a summary of cars, but you can't see that in the output (so you'll have to trust me!). This is useful for designing tutorials where you don't want to show the answers straight away. To use this parameter you would type: {r setup, include=FALSE} at the beginning of the chunk. If you want to include data in the output, you should also put the data here too: For example, a chunk with {r setup, car, include=FALSE} and then summary(cars) would not show anything - as the settings have been set to not display any code or results. If include=TRUE it will display all code and results in the code chunk.

Below is an example of a chunk of code used at the beginning to set up the document it includes the {r ,include = T} statement, so you can see the code. Setup refers to global applied things such as the dataset cars. R will then import the dataset and use it through, without the need to constantly call this dataset.

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(engine.path = list(python = 'C:/Users/AT003502/Anaconda3/python.exe'))
#tells R where to look at for the Python console
```

Below is an exmple where the argument {r, include = F} has been used, so you can only see the output and not the code used to generate the output. You can always check the code by the going to the raw rmd (r markdown) script if you do not want to display the code.

2. {r eval = FALSE} {r eval = TRUE}

Here's some code using {r eval=FALSE}. There are a number of default parameters you don't need to state. For example echo = TRUE is the default, so there is no need to include it in the curly braces. {eval=FALSE} displays the code, but does not display the output of the code - for example the plot and summary data are not shown.

```
summary(AirPassengers)
str(AirPassengers)
series<-ts(AirPassengers)
plot(series)
```

Here is an example using eval=TRUE (don't forget, echo = TRUE as a default here). You can see the code and the output below the code.

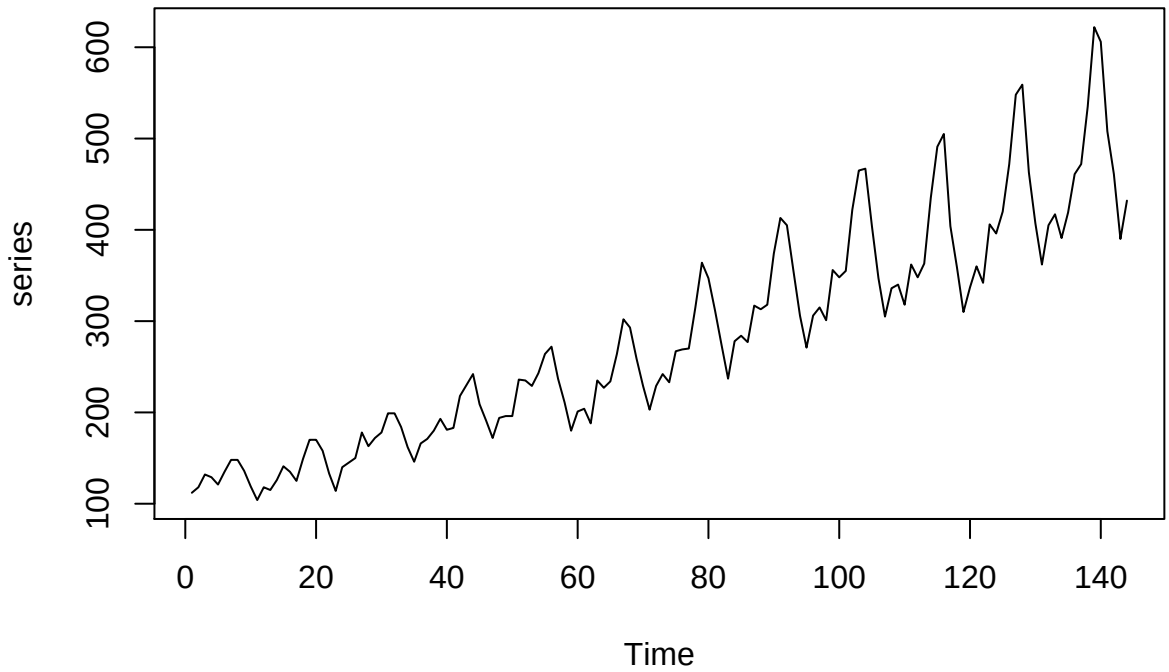
```
summary(AirPassengers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   104.0   180.0   265.5   280.3   360.5   622.0
```

```
str(AirPassengers)
```

```
##  Time-Series [1:144] from 1949 to 1961: 112 118 132 129 121 135 148 148 136 119 ...
```

```
series<-ts(AirPassengers)
plot(series)
```

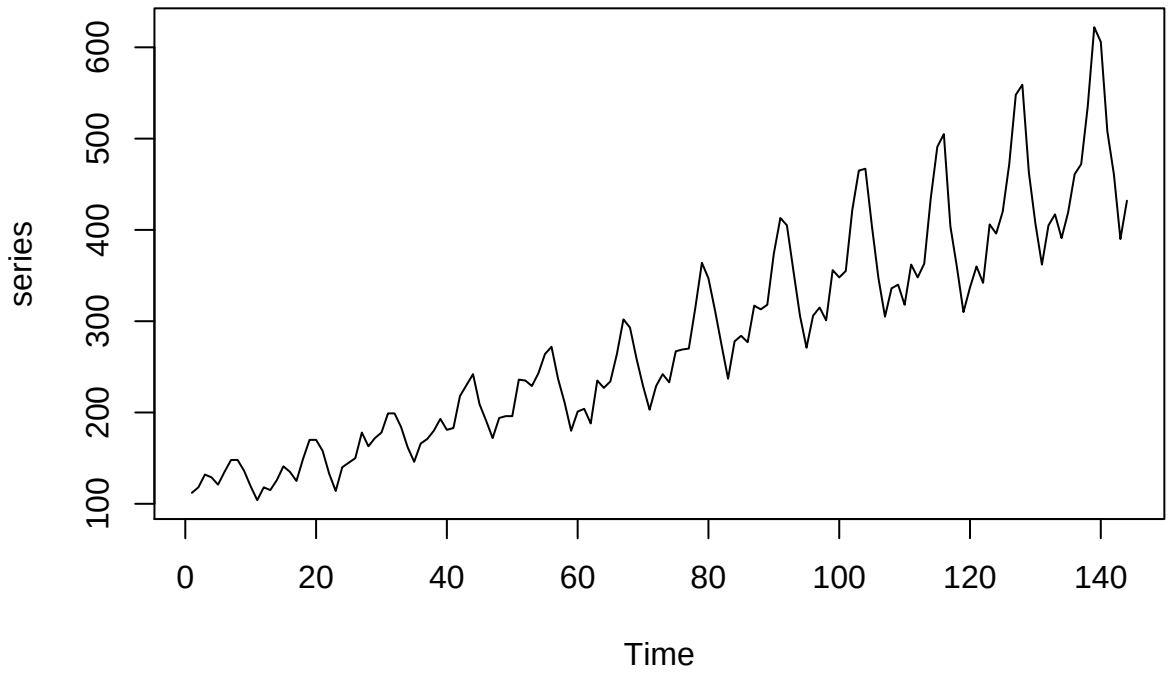


4. {r echo = FALSE}

Here is an example using echo = F (default for eval = TRUE, so no need to state this inside the curly braces).Now you can only see the results, but not code used to generate the output. Useful for displaying data and plots only. Here we've selected to show the summary of the data for AirPassengers and the whole dataset.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   104.0   180.0   265.5   280.3   360.5   622.0
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```



2. Formatting text

There are a number of different options for formatting e.g. headings, bold, italtics, equations etc. This is for text *not* included inside code 'chunks'. Examples of text formating:

- New paragraph: End a line with two spaces

- Italics: one asterisk on either side of word or sentence produces *italics*
- Bold: one asterisk on either side of word or sentence produces **bold**
- Bullet Points: Start line with asterisks and space before the text (*). Ensure to leave a space of one line between each bullet point. Leave two spaces at the end of the bullet point to ensure new bullet point starts.
  - Subpoints: To follow a bullet point with a subpoint, go to the next line and press tab key twice. Use a plus and space (+) before typing text.
- Headers are achieved by using the hash key (#) followed by a space, one # represents the largest header, two (##) represents header two etc:

Header 1 example

Header 2 example

- In line R code can be performed using a backtick, followed by r and closed with a backtick. E.g. There were 50 cars studied uses the argument nrows(cars) to calculate the number of cars.

1. Ordered lists are achieved by using numbers, followed by a space instead of an asterisks.
2. The same format for subpoints applies as before
   - using two tabs and the plus sign to denote the subpoint

- Tables are also constructed usings dashes (-) under heading titles and bars (|) to separate columns:

header1	header 2
info	info 2
infor 3	infor 4
this is	great

Links can also be added by defining the name for the link in square brackets link:, followed by the link and "Title". This Stores the link as the word used in square brackets(e.g. Link would appear here (instead of square brackets) [information][id]).

This is the link

3. Other languages

R Markdown can utilise languages other than R - for example Python, SQL and JavaScript :

Python

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

```
## ['hello,', 'python', 'world!']
```

Note: you need to set the engine path to the programmes if you are not running these languages in Linux. For example, at the beginning of the document the path for Python is given:

knitr::opts_chunk$set(engine.path = list(python = 'C:/Users/AT003502/Anaconda3/python.exe'))

This directs R to the Python.exe programme.