

# An introduction to R markdown

*C Wall*

*9/18/2018*

## This is an R Markdown!

R markdown is great, and when I say great, **I mean REALLY great.**

When exporting your markdown you can... (1) *embed figures*

(2) *view tables*

(3) *export models*

(4) *archive your data pipeline*

You can write code and execute functions in a similar way to normal R script files. However, there are a few distinct differences between R scripts and R markdown scripts. Some of these differences relate to how code is executed. Other differences can be seen in the added benefit of integrating scripts with outputs and other embedded properties.

The benefit of using markdown with colleagues is they can actually see the script, the figures, the path of model selection, even the assumptions of ANOVA (QQ plots, etc) that normally are hidden behind the curtain. In short, **R markdown helps make your science repeatable, sharable, and data analysis coherent!**

Here are some common commands and examples of what Markdown can do for you. I recommend visiting the **R Bookdown** for a complete and exhaustive guide <https://bookdown.org/yihui/rmarkdown/> or the R markdown cheat sheet

## Markdown basics

### Common Commands

*Notice the script at the very top of your markdown (this is not standard, but is customized)*

output:

html\_document:

code\_folding: hide

toc: yes

toc\_depth: 4

toc\_float: yes '

Let's go through this line by line...

html\_document is an option you originally set when you open the markdown. This is how your file will be exported.

code\_folding: hide will allow you to *show* or *hide* all code (option at top of file)

toc is for table of contents on the left

toc\_depth is how long you will have table of contents before folding over

toc\_float lets your contents move as you advance in your document.

*Play around with these options, and see how they affect your code*



Figure 1: Pa'āni is *beautiful* (and **sassy**)

### The setup chunk

`r` **setup** This is your set up code and is a useful way to get around the fact that knitr will look for your .Rmd file and all files in the this directory. By setting the root.dir you can force knitr to look for files in the directory you specify and the folders within.

In the setup chunk (the first code chunk) you can set ‘global’ options, such as message/warning exclusion, or hiding results or outputs.

`knitr::opts_chunk$set(warning=FALSE, message=FALSE)` This command here is requiring the package `knitr` in the code chunk (‘set’ for set up) and is saying to “make all warning and message FALSE” i.e., hide them from output.

`include=FALSE` This command makes your code absent from final html. It is executed but the code is hidden.

`eval=FALSE` This command allows you to show the code in your script within R studio, but not evaluate (or run) the code. It is effectively silent in your analysis and output html.

`echo=FALSE` This will show the output but not the code chunk... notice the difference.

`results='hide'` This will hide all results in a code chunk, or any returned results from your commands.

`fig.show=FALSE` This will hide the figures you generate. This may be useful for some of those assumptions of ANOVA– you want to see it, but do you want all of them in your final archived datafile? Maybe not.

`fig.height=3, fig.width=3` This will format your figure to be 3x3

### Text formatting

It is useful to understand how to modify text and format headings. You can do this in a variety of ways.

A line break—is executed by two spaces at the end of previous line, and a return  
use # for headings, #.... ##### and so on. # is largest and ##### smallest heading

Two forms for *italics* and *italics* and **bold** and **bold** to modiy text

Superscript superscript<sup>2</sup> or strikethroughs ~~strikethrough~~

add links R studio link

endash: – emdash: — and ellipsis: ...

inline equation:  $A = \pi * r^2$  or  $y = a + b * x$

(a line can inserted with \*\*\*)

### Code chunks

```
## love your data and it will love you back
getwd()
```



Figure 2:

```
## [1] "/Users/chriswall/Desktop/zenodo/Intro to Markdown"
```

Code chunks are where your code is executed. If you do not set the working directory in `setup` each code chunk will revert to its original directory, or where your .Rmd file lives. Below: This is a code chunk, and this is how you enter your data into R markdown. Note the `code chunks` always start with a ````{r ...}` and ends with a `...}````

The code chunk below is an example of attaching the data (.csv), familiar to you R-heads. Since the data file is in the directory specified by `knitr::opts_knit$set(root.dir = ...)` above, you can reference the .csv easily.

```
#####
# import data, observe structure
#####

# data file is in the folder 'data', within main working directory
data<-read.csv("data/coral_data.csv")
names(data)

##  [1] "Time.point"    "Period"       "Site"        "Species"     "Status"
##  [6] "Sample.ID"     "Pair"        "Depth.ft"    "biomass"     "chl.a"
```

## Including figures

You can include figures from script output as results, or figures from files in your directory. First, let's see how you can add an image from a file to your markdown.

**Figure 1.** This is a figure of a bleached *Montipora capitata* coral during the 2014 bleaching event in Kāne'ohe Bay, O'ahu, Hawai'i. (PC: CB Wall)

In this chunk we will make a figure of chlorophyll *a* concentrations cm<sup>-2</sup>. Some of the syntax here should be familiar.

```
# message = FALSE to hide any messages with exporting the figure with dev.off(), or anything else that ...

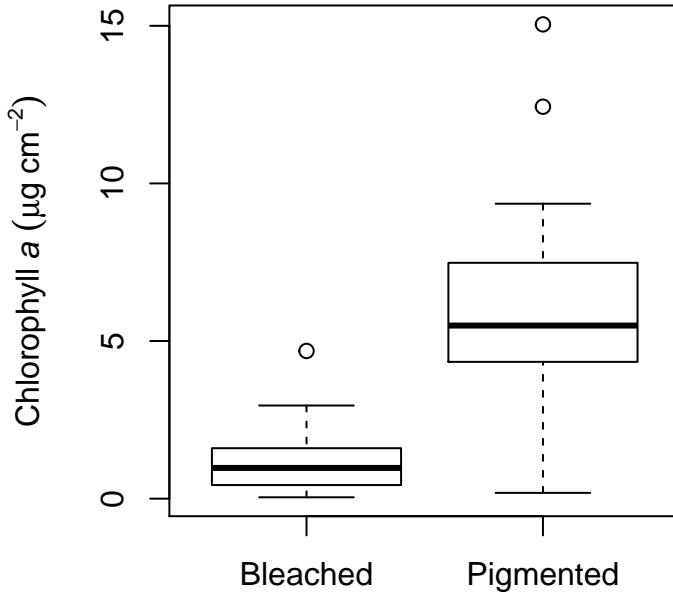
par(mfrow=c(1,1), pty="sq")
Status.label<-c("Bleached", "Pigmented") # for x labels
```

```

plot(chla~Status, data=data, xaxt="n",
      ylab=expression(paste("Chlorophyll", ~italic("a"), ~(mu*g~cm^-2), sep="")),
      xlab="Coral Physiological State",
      main="Example Figure")
axis(1, at=1:2, labels=Status.label) # this plots new labels on the x axis (axis = 1)

```

## Example Figure



```

##### save the figure and export to directory? #####
# in this case, there is a file in my directory names 'figures' where I want plots to go

dev.copy(pdf, "figures/Chlorophyll.pdf", encod="MacRoman", height=5, width=5)
# height and width set here for the output figure
dev.off() # close the object

```

If you change export to the same location as your working directory and not a subfile “chlorophyll.pdf“ will do the job.

## Examples

- see a code chunk in my markdown file, but it isn't executed if `eval=FALSE` is set

```
mean(data$chla)
```

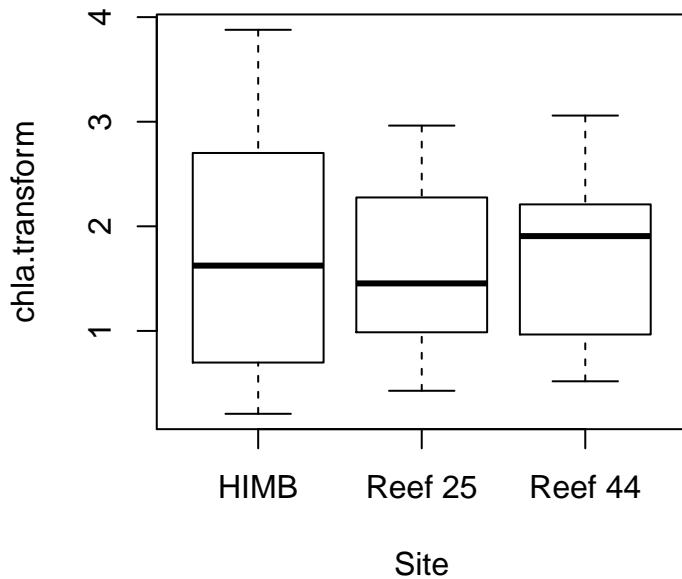
- execute the code and hide results `results='hide'`

```
chla.transform<-sqrt(data$chla)
mean(chla.transform)
```

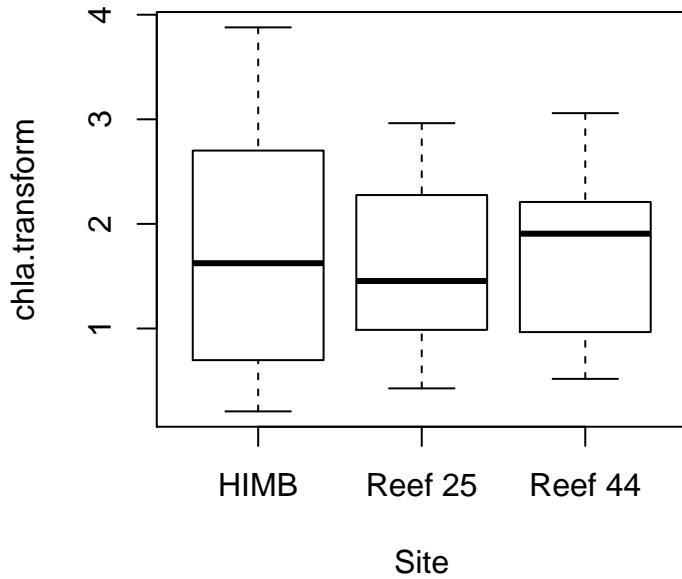
- execute and show code and SEE the figure I generate, but hide results (returned data)  
`results='hide', fig.height=4, fig.width=4, fig.align='center'`

```
mean(chla.transform)
```

```
# make figure 4 x 4 and center align  
plot(chla.transform~Site, data=data)
```



... or just print the figure with echo=FALSE sans the code



```
... or show NOTHING with results='hide', fig.show=FALSE
```

```
plot(chla.transform~Site, data=data)
```

Make a table of summary data

plot raw data ‘as is’

```
knitr::kable(data[c(1:5), c(1:10)])
```

Time.point	Period	Site	Species	Status	Sample.ID	Pair	Depth.ft	biomass	chl.a
2014 Oct	Bleaching	HIMB	MC	B		3	2	0.6666667	21.22296
2014 Oct	Bleaching	HIMB	MC	NB		4	2	0.6666667	27.31830
2014 Oct	Bleaching	HIMB	MC	B		5	3	1.0000000	19.74599
2014 Oct	Bleaching	HIMB	MC	NB		6	3	1.0000000	15.44902
2014 Oct	Bleaching	HIMB	PC	B		9	5	1.6666667	26.27286

Or make summary data and include this table in your markdown

```
chl.mean<-aggregate(chla~Time.point + Site + Species + Status, data, mean)
biomass.mean<-aggregate(biomass~Time.point + Site + Species + Status, data, mean)
data.table<-cbind(chl.mean[, c(1:5)], biomass.mean[,5])
colnames(data.table)<-c("Time Point", "Site", "Species", "Status", "mean chla", "mean AFDW")
knitr::kable(data.table)
```

Time Point	Site	Species	Status	mean chla	mean AFDW
2014 Oct	HIMB	MC	B	1.4407045	19.38053
2014 Oct	Reef 25	MC	B	1.1771529	16.72889
2014 Oct	Reef 44	MC	B	1.9240275	21.35410
2014 Oct	HIMB	PC	B	0.6808416	30.96592
2014 Oct	Reef 25	PC	B	1.1561949	28.72490
2014 Oct	Reef 44	PC	B	0.8367104	38.47826
2014 Oct	HIMB	MC	NB	5.1866494	27.21420
2014 Oct	Reef 25	MC	NB	3.6726268	26.55296
2014 Oct	Reef 44	MC	NB	4.5428889	28.49259
2014 Oct	HIMB	PC	NB	10.3363654	37.18556
2014 Oct	Reef 25	PC	NB	6.3499443	47.68068
2014 Oct	Reef 44	PC	NB	7.3410721	42.94596

## Running models

markdown makes running models easy. You can leave all candidate models in the script or you can only report final models. In either case, it is an easy way to keep your data from analysis easy to understand and to QA/QC before publication

**chlorophyll a model** Load packages and look at structure

```
library("lme4")
library("lmerTest")
library('effects')

str(data)
data$Sample.ID<-as.factor(data$Sample.ID)
```

Run a linear mixed effect model, see the anova output, random effects, and plot effects

```
mod<-lmer(chla~Species*Site*Status+(1|Pair), data=data)
anova(mod, type=2)
```

```
## Analysis of Variance Table of type II with Satterthwaite
## approximation for degrees of freedom
##             Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)
## Species      30.28   30.28     1     47  10.566  0.002133 **
```

```

## Site          17.52    8.76    2    47   3.056  0.056521 .
## Status       369.23  369.23    1    47  128.814 4.663e-15 ***
## Species:Site 4.46     2.23    2    47   0.778  0.465048
## Species:Status 63.93   63.93    1    47  22.304 2.134e-05 ***
## Site:Status  21.98   10.99    2    47   3.833  0.028692 *
## Species:Site:Status 6.59    3.29    2    47   1.149  0.325777
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
rand(mod)

## Analysis of Random effects Table:
##      Chi.sq Chi.DF p.value
## Pair 2.84e-14      1      1
plot(allEffects(mod), ylab="total chlorophyll/cm2")

```

### Species\*Site\*Status effect plot

