josh does a markdown

In general, Rmarkdown separates sections via three `.

* these sections are known as ‘code chunks’
  + they organise your document for easier reading
    - you can quickly insert chunks by pressing **ctrl + alt + I**
    - alternatively, you can add chunks with the add chunk button (green c with a plus)

For annotations you don’t want read, leave them out of code chunks.

There are many formatting options available in markdown including

* *italics*
* **bold**
* code
* you may also want to produce asubscript or asuperscript

In addition, you can make heading levels by adding # at the front of the line, these headings can also have in-line formatting;

# *hi* **there**

## *hi* **there**

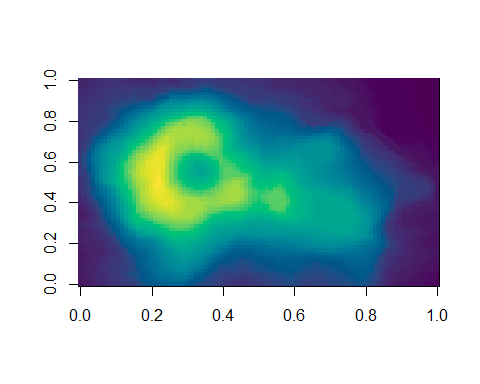
### *hi* **there**

To produce the file you have been working on in Rmarkdown either click the knit button, or use **ctrl + shift + k**

For a detailed list of formatting please see this [link](https://pandoc.org/MANUAL.html#pandocs-markdown)

Rmarkdown provides an excellent method to integrate text into writing scripts, from simply annotating code for easier reading, up to producing documents and powerpoint presentations.

Images and figures are integrated straight into the document, and code can be run directly in the script to display the output while you work.



The Maunga Whau volcano

You can create, and alter dataframes much the same as you would in Rstudio

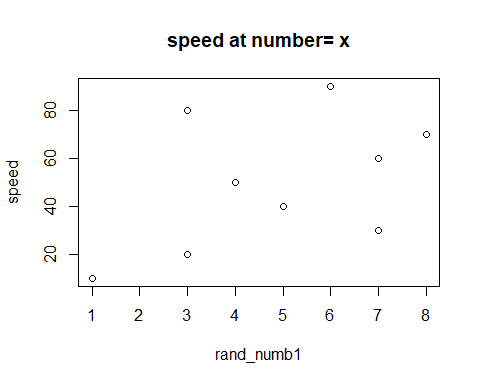
rand\_numb1 <- c(1, 3, 7, 5, 4, 7, 8, 3, 6)  
speed <- c(10, 20, 30, 40, 50, 60, 70, 80, 90)  
Rmkd\_dataset <- data.frame(rand\_numb1, speed)  
head(Rmkd\_dataset)

## rand\_numb1 speed  
## 1 1 10  
## 2 3 20  
## 3 7 30  
## 4 5 40  
## 5 4 50  
## 6 7 60

summary(Rmkd\_dataset)

## rand\_numb1 speed   
## Min. :1.000 Min. :10   
## 1st Qu.:3.000 1st Qu.:30   
## Median :5.000 Median :50   
## Mean :4.889 Mean :50   
## 3rd Qu.:7.000 3rd Qu.:70   
## Max. :8.000 Max. :90

plot(Rmkd\_dataset, type = "p", main = "speed at number= x")



The knitr function also has functionaility in a variety of other porgramming languages, including; + Python + SQL + Bash + Rcpp + Stan + JavaScript + CSS

All that is necessary is to change the **r** preface at the start of a code chunk

For example #{bash} #

#{python} #

etc.

The advantage of this is that it allows integration of the script across a multitude of different coding langauges. This can be useful, for example if you have specific processes that requires programs not available in R + For example, in my case I used many program suites not available in R for data preparation and filtration. + At the time I had to maintain multiple different scripts across the programs to handle this, but had I used Rmarkdown it would have allowed me to contain it all within the single script.