Wonderful markdown

Wonder Woman

Table of Contents

# A Title

Make text **bold** or *italic* or code

* First thing to do
* Second thing to do
* Third line
* make sure you leave a blank line so lists will appear as new paragraph
* style of dot point in markdown doesn’t change on html output
* OK?

Some text that needs footnoting[[1]](#footnote-21)

1. numbered list
2. example
3. all good

[Link text here](https://csiro-data-school.github.io/rmarkdown/01-Literate-Programming/index.html)



“image caption - my 30 years cake”

## A Subtitle

Add a sentence in your document that shows the versions of the software used to create it. Using inline code sections, describe which version of R and the tidyverse package you are using by calling the version of R, which is 3.6.1 and the R package tidyverse version is 1.2.1.

# everything between the back-ticks is now recognised not as text but as code  
  
6\*9

## [1] 54

## A Second

### A Section Heading

writing R code into the document

library(tidyverse) # load (and run) teh tidyverse

## -- Attaching packages ---------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.2  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

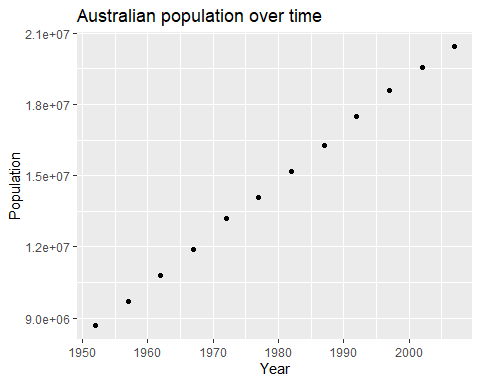
gapminder <- read\_csv("data/gapminder.csv") # create the gapminder variable

## Parsed with column specification:  
## cols(  
## country = col\_character(),  
## continent = col\_character(),  
## year = col\_double(),  
## lifeExp = col\_double(),  
## pop = col\_double(),  
## gdpPercap = col\_double()  
## )

australia <- gapminder %>%   
 filter(country == "Australia") #filter for Australia  
  
australia # prints the australia dataframe to screen

## # A tibble: 12 x 6  
## country continent year lifeExp pop gdpPercap  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Australia Oceania 1952 69.1 8691212 10040.  
## 2 Australia Oceania 1957 70.3 9712569 10950.  
## 3 Australia Oceania 1962 70.9 10794968 12217.  
## 4 Australia Oceania 1967 71.1 11872264 14526.  
## 5 Australia Oceania 1972 71.9 13177000 16789.  
## 6 Australia Oceania 1977 73.5 14074100 18334.  
## 7 Australia Oceania 1982 74.7 15184200 19477.  
## 8 Australia Oceania 1987 76.3 16257249 21889.  
## 9 Australia Oceania 1992 77.6 17481977 23425.  
## 10 Australia Oceania 1997 78.8 18565243 26998.  
## 11 Australia Oceania 2002 80.4 19546792 30688.  
## 12 Australia Oceania 2007 81.2 20434176 34435.

ggplot(data = australia,  
 mapping = aes(x = year,  
 y = pop)) +  
 geom\_point() +  
 labs(title = "Australian population over time",  
 x = "Year",  
 y = "Population")



### Another section

This is to show how the code can be done in “chunks” but KnitR will recognise all the code in the document as one; so you can refer to data previously created. Tip: if you have long line of code, you can select a section, and go to the insert button. I am copying the code above to demonstrate this and then splitting it up with comments

We are also going to add some options for how the code and or output will display. These are inside the curly brackets. the options are available form the settings button (cog image) in the code chunk

eval=FALSE will not run yuur data, it will show the code but not ourput echo=FALSE controls if the code is showed in the document (and only show the output) results whether or not to show the results at that point warning whether or not tos show warnings message whether or not to show messages

This is the first chunk to load the tidyverse, displayed without messages and warnings just the code line

library(tidyverse) # load (and run) the tidyverse

This is the second chunk. Read in the data, filter for Australia, display the Australia dataframe.

There are 1704 rows in the gapminder data.

gapminder <- read\_csv("data/gapminder.csv") # create the gapminder variable

## Parsed with column specification:  
## cols(  
## country = col\_character(),  
## continent = col\_character(),  
## year = col\_double(),  
## lifeExp = col\_double(),  
## pop = col\_double(),  
## gdpPercap = col\_double()  
## )

you usually run the code so you can get some parameters to refer to in the following paragraph

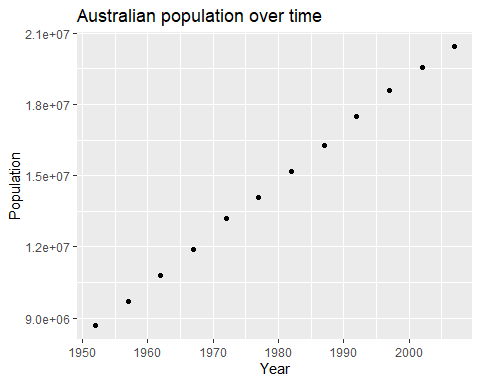
australia <- gapminder %>%   
 filter(country == "Australia") #filter for Australia  
  
australia # prints the australia dataframe to screen

## # A tibble: 12 x 6  
## country continent year lifeExp pop gdpPercap  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Australia Oceania 1952 69.1 8691212 10040.  
## 2 Australia Oceania 1957 70.3 9712569 10950.  
## 3 Australia Oceania 1962 70.9 10794968 12217.  
## 4 Australia Oceania 1967 71.1 11872264 14526.  
## 5 Australia Oceania 1972 71.9 13177000 16789.  
## 6 Australia Oceania 1977 73.5 14074100 18334.  
## 7 Australia Oceania 1982 74.7 15184200 19477.  
## 8 Australia Oceania 1987 76.3 16257249 21889.  
## 9 Australia Oceania 1992 77.6 17481977 23425.  
## 10 Australia Oceania 1997 78.8 18565243 26998.  
## 11 Australia Oceania 2002 80.4 19546792 30688.  
## 12 Australia Oceania 2007 81.2 20434176 34435.

We can now see there are 12 rows in the australia data

and this is the last chunk to plot the data

ggplot(data = australia,  
 mapping = aes(x = year,  
 y = pop)) +  
 geom\_point() +  
 labs(title = "Australian population over time",  
 x = "Year",  
 y = "Population")



or to load global settings - HINT using the double colon :: will allow you to load specific parts of a package without having to load the package, eg if we were to use the code in an R chunck

knitr::opts\_chunk$set(message=FALSE, warning=FALSE, echo=FALSE, fig.width=11) it will set all messages and warnings to false (display code only) and set the figures to a standard width 11 inches

1. This is the footnote text [↑](#footnote-ref-21)