R Markdown Tips and Testing

Your Name (your-uni)

2021-09-23

```
#Load libraries
library(tidyverse)
library(gapminder) #only for this example
```

I can write normal text here.

```
#I can also write text as comments within a code chunk.
#The shortcut is cmd+alt+i (or ctrl+alt+i in windows).
```

Here is how numbering works.

1 Heading

1.1 Subheading

1.1.1 Sub-subheading

I can suppress numbering, by adding "{-}"

Suppress numbering here.

Here is some inline code: two plus two equals 4.

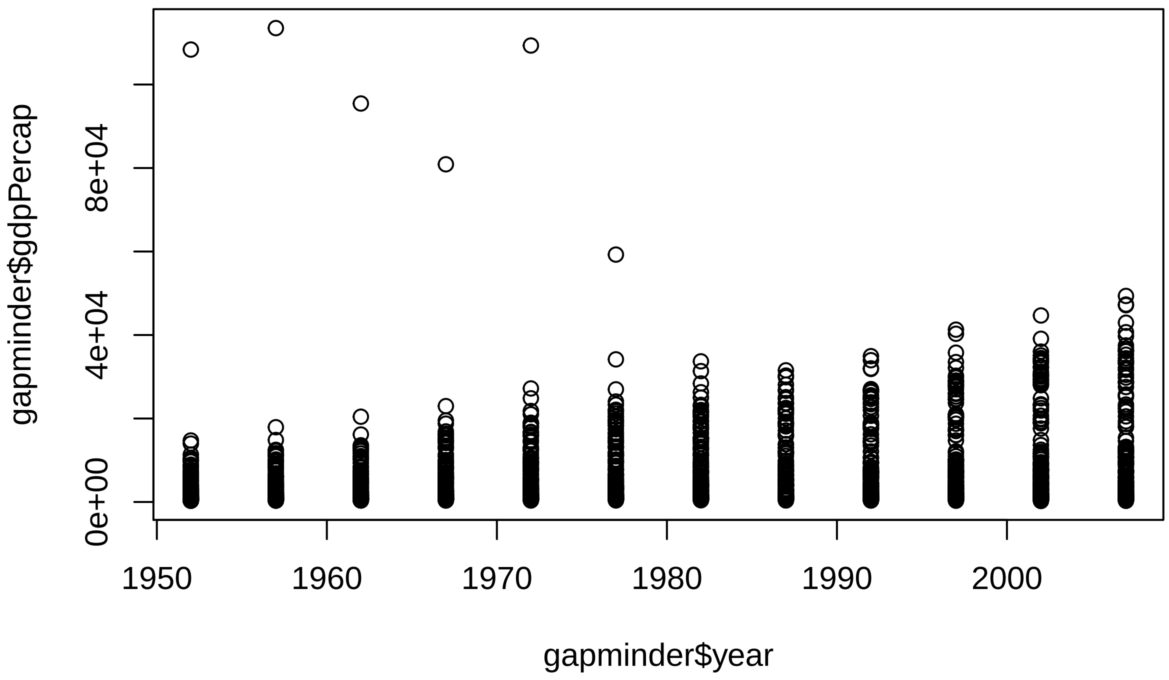
Here is a code chunk, displaying a table and the code used to generate it. Piping a dataframe to knitr::kable makes it look nicer in rmarkdown.

```
gapminder %>%
  head(5) %>%
  knitr::kable()
```

country	continent	year	lifeExp	pop	gdpPercap
Afghanistan	Asia	1952	28.801	8425333	779.4453
Afghanistan	Asia	1957	30.332	9240934	820.8530
Afghanistan	Asia	1962	31.997	10267083	853.1007
Afghanistan	Asia	1967	34.020	11537966	836.1971
Afghanistan	Asia	1972	36.088	13079460	739.9811

Here is another code chunk, displaying a plot and the code used to generate it.

```
plot(gapminder$year, gapminder$gdpPercap)
```



Let's make a new page.

Code chunk display options

In the setup chunk we defined the default (global) code chunk display option to be echo=TRUE, meaning the knitted chunk will display both code and results.

Note that the global settings also suppress warnings and messages. Usually, you will not have to change these settings when answering assignment questions.

For example, suppose I want to display the structure of the gapminder dataset.

- Use **{r}** for the default options

```
str(gapminder)
```

```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##  $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163
##  $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

```
#default display, inherited from the setup chunk.
#this is how to display answers for most assignment questions.
```

If we want, we can change the display options within the code chunk. Suppose I only want to display the results, and not the code.

- Use "**{r echo = FALSE}**"

```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##  $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163
##  $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

And the reverse, display the code (and not the results)

- Use "**{r eval = FALSE}**"

```
str(gapminder)
```

Or, execute and not display anything (neither code nor results)

- Use "**{r include = FALSE}**"

(nothing here)

Additional tips

Don't include "clunky" output that is not integral to answering the question.

As an example, consider the following question: *what is the mean GDP per capita of the gapminder dataset?*

The following is not a very good approach, since the output is clunky.

```
summary(gapminder)
```

```
##         country        continent        year         lifeExp
##  Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
##  Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
##  Algeria    :  12   Asia    :396   Median :1980   Median :60.71
##  Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
##  Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
##  Australia  :  12                  Max.   :2007   Max.   :82.60
##  (Other)    :1632
##       pop               gdpPercap
##  Min.   :6.001e+04   Min.   :   241.2
##  1st Qu.:2.794e+06   1st Qu.:  1202.1
##  Median :7.024e+06   Median :  3531.8
##  Mean   :2.960e+07   Mean   :  7215.3
##  3rd Qu.:1.959e+07   3rd Qu.:  9325.5
##  Max.   :1.319e+09   Max.   :113523.1
##
```

```
#by inspection, 7215.3
```

The following is neater, but also not very good, since it still displays unecessary information, and the answer is "hard coded" - i.e. numbers are typed in - this is bad practice.

```
summary(gapminder$gdpPercap)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    241.2   1202.1   3531.8   7215.3   9325.5 113523.1
```

```
#by inspection, 7215.3
```

The best way is to use inline code, as per below:

The mean GDP in the gapminder dataset is 7215.3.

In general:

- Do not hard code (i.e. do not type numeric values directly)
- Instead, use inline code for simple calculations
- And use code chunks for more complicated/longer operations
- Be sensible about what output to display, and what to omit

Some final remarks

Remember, you can execute code in rmarkdown without knitting, just like in a .r script.

Since knitting is memory exhaustive, in general, try to knit only when you're confident your code works, and/or when you want to view the formatting of the final output.

For more detail on rmarkdown visit Yihui Xie's website.