Simple process for sharing R code via Docker

Ronald (Ryy) Glenn Thomas

2024-01-29

Table of contents

1 Introduction

Lets assume you have an rmarkdown `Rmd` file, say `peng.Rmd`, that you're written to analyze some data. You now want to share the code with a colleague, we'll call him Joe. How to proceed?

The simplest option is simply to send Joe the "rmd" file containing the code via the most convenient method (e.g. email/text/slack/discord/github/USB drive etc.)

The next step will be for the Joe to (attempt to) load and run the code. Typically he would do this with either using Rstudio.app`to open the file and`knit‘ it, render it from the command line with something like

```
> R -e "render('script.Rmd')"
```

Sometimes this approach works, and all is well. Joe can add comments or expand the code and reply to you. Frequently, however, the process will fail for any number of reasons. Ideally to facilitate reproducibility Joe will have as similar a computing environment as you, the original developer. This can be difficult to achieve, especially given the dynamic nature of open source software. For example Joe may have an outdated version of `R` installed on their workstation, or their `R` environment may be missing a necessary package. Additiional potential problems include: the package may be present but its the wrong version, the program may need to source an additional file thats missing, or the program load some data that it can't find on Joe's machine.

All of these problems go away if instead of sending the program as a standalone text file you send it as a docker image. In this post we'll walk through the process of dockerizing the R code.

````
---
title: "Penguins analysis"
author: "R.G. Thomas"
date: "`r format(Sys.time(), '%B %d, %Y')`"
fontsize: 11pt
geometry: "left=3cm,right=5cm,top=2cm,bottom=2cm"
output:
  pdf_document:
    keep_tex: true
    includes:
      in_header: "~/shr/preamble.tex"
---

```{r include=F, echo=F}
library(pacman)
p_load(palmerpenguins, tidyverse, knitr)
opts_chunk$set( warning = FALSE, message = FALSE, echo = FALSE, results =
           "asis", dev = "pdf")
```

# Introduction

We can work with the dataset `penguins` included in the package
````

````
`palmerpenguins`.  One naive approach is to split the dataset and do three
separate analyses:

```{r }
df1 <- split(penguins, penguins$species)

foo <- function(df, z) {
  df |> ggplot(aes(x = bill_length_mm, y = flipper_length_mm)) +
    geom_point(aes(color = island), alpha = .5) +
    geom_smooth() +
    scale_color_manual(values = c("purple", "green", "red")) +
    theme_bw() +
    labs(
      title = paste(z, " Penguin Anatomy Comparison"), x = "Flipper length",
      y = "Bill length", color = "Island"
    )
  plotfile_name <- paste0(z, ".pdf")
  ggsave(plotfile_name)
  cat(paste0("\\includegraphics[height=3cm]{", plotfile_name, "}"), "\n")
  cat("\\vspace{1cm}", "\n")
}

bar <- df1 |> map2(names(df1), foo)
```
````

The Rmd file runs cleanly and generates the following page. However, we note that the third plot needs additional examination and want to relay the program to our colleague Joe for further analyis.

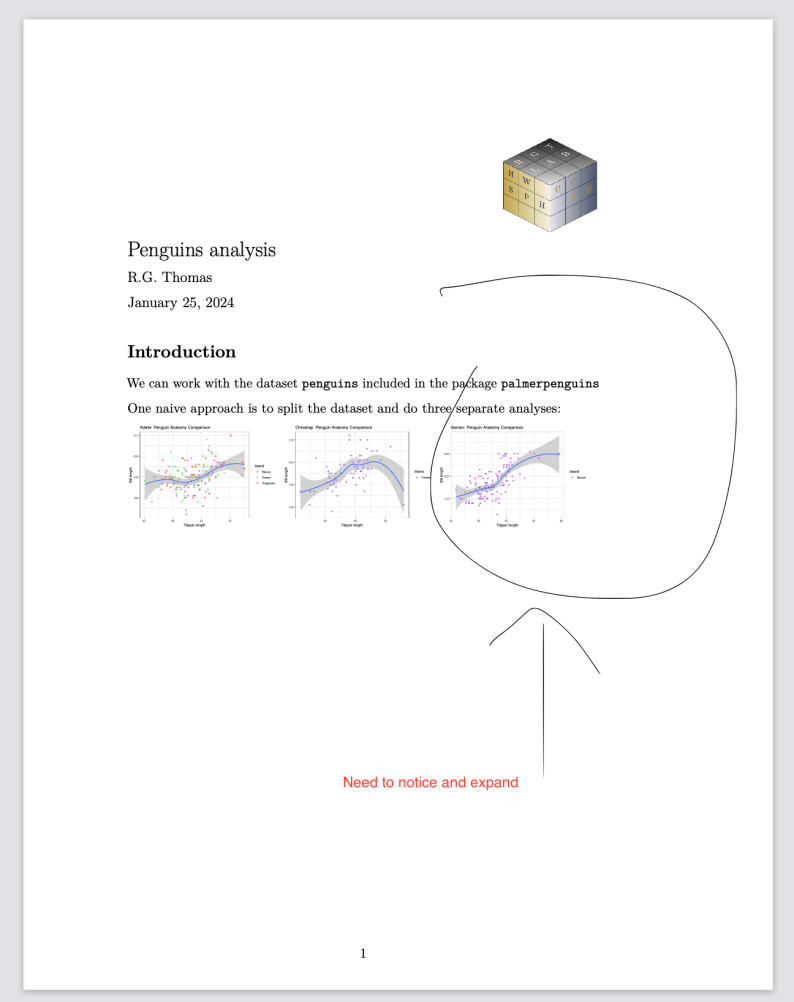

Figure 1: rendered page

Whats the best way to accomplish this?

We start by simply emailing the file to him (rgthomas4747@gmail.com) and asking

him to collaborate.

Joe downloads the attachment. Opens a working directory and attempts to run the Rmd file to see whats what.

Joe has a linux mint desktop

```
> mkdir peng_collaboration
> cd peng_collaboration
> R -e "render('peng.Rmd')"
```

Linux can'f find R

Joe can fix this by installing R > sudo apt install r-base-core

The first problem is that R can not find the function `render`.

He determines that render is a function in the package rmarkdown

He tries to install rmarkdown with R -e "install.packages('rmarkdown')"

This fails due to inadequate permission on the directory /usr/lib/R/library

After some research he decides to install Rstudio to shortcut some of these problems.

Now try again > R -e "library(rmarkdown); render('peng.Rmd')"

Comes back with an error saying no `pacman` package.

R -e "install.packages('pacman')"

This is better but something is taking a really long time to install...

Tidyverse is breaking on install. Looks like its missing some libraries. Some research leads to the suggestions:

```
sudo apt install libssl-dev libcurl4-openssl-dev unixodbc-dev libxml2-dev\
libmariadb-dev libfontconfig1-dev libharfbuzz-dev libfribidi-dev\
libfreetype6-dev libpng-dev libtiff5-dev libjpeg-dev
```

One more try... and the latex engine notes the absence of the file

```
~/shr/preamble.tex
```

So, I need to relay the missing .tex file.

Also the .pdf (sudoku.pdf) logo file.

Finally! success. # Docker approach

Alternatively, consider the “Docker” approach.

Before sending peng.Rmd to Joe we’ll dockerize it.

- Prepare a work directory: penguins. We want to send Joe a container that has R and all the preliminaries taken care of so that all he has to do is

Here is the docker file

```
---
title: "Penguins analysis"
author: "R.G. Thomas"
date: "January 29, 2024"
fontsize: 11pt
geometry: "left=3cm,right=5cm,top=2cm,bottom=2cm"
output:
  pdf_document:
    keep_tex: true
    includes:
      in_header: "preamble.tex"
---



# Introduction

We can work with the dataset `penguins` included in the package `palmerpenguins`.




One naive approach is to split the dataset and do three separate
analyses:
```

```
\includegraphics[height=3cm]{Adelie.pdf}
\vspace{1cm}
\includegraphics[height=3cm]{Chinstrap.pdf}
\vspace{1cm}
\includegraphics[height=3cm]{Gentoo.pdf}
\vspace{1cm}
```

run docker

```
docker build -t penguin_review --platform=linux/amd64 .
```

```
docker push
```

relay image to Joe

```
> docker run -it --rm rgt47/penguin_review
```

```
> R -e "library(rmarkdown); render('peng.Rmd')"
```

and if he wants to edit peng.Rmd

```
> vim peng.Rmd
```

```
\usepackage[export]{adjustbox}
\usepackage{fancyhdr}
\usepackage{titling}

\pagestyle{fancy}

\pretitle{
\begin{flushright}
\includegraphics[width=3cm,valign=c]{sudoku.png}\\
\end{flushright}
\begin{flushleft} \LARGE }
\posttitle{\par\end{flushleft}\vskip 0.5em}
\predate{\begin{flushleft}\large}
```

```
    \postdate{\par\end{flushleft}}
    \preauthor{\begin{flushleft}\large}
    \postauthor{\par\end{flushleft}}
\fancyfoot[L]{\currfilename} %put date in header
\fancyfoot[R]{\includegraphics[width=.8cm]{sudoku.png}}
\fancyhead[L]{\today} %put current file in footer
```

2 REFERENCES

Running your R script in Docker