

Simple process for sharing R code via Docker

Ronald (Ryy) Glenn Thomas

2024-01-25

Table of contents

1 Introduction

1

1 Introduction

Lets assume you have an `Rmd` file, say `script.Rmd`, you're written to analyze some data. You now want to share the code with a colleague. How to proceed?

The simplest option is simply to send him/her the “rmd” file containing the code via email/text/slack/discord/USB drive etc.

The next step will be for the code recipient to attempt to load and run the code typically using either of two methods 1) use `Rstudio.app` to open the file and `knit` it, or 2) render it from the command line with something like

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

Sometimes this works. Frequently however the process will fail for any number of reasons. Ideally to facilitate reproducibility the new user will have as similar a computing environment as



the original developer. This can be difficult to achieve especially given the dynamic nature of open source software. For example your colleague may have an outdated version of R installed on their workstation, or their R environment may be missing a necessary package, or the package may be present but its the wrong version, or the program may need to source an additional file thats missing, or load some data that it can't find.

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)
```

```

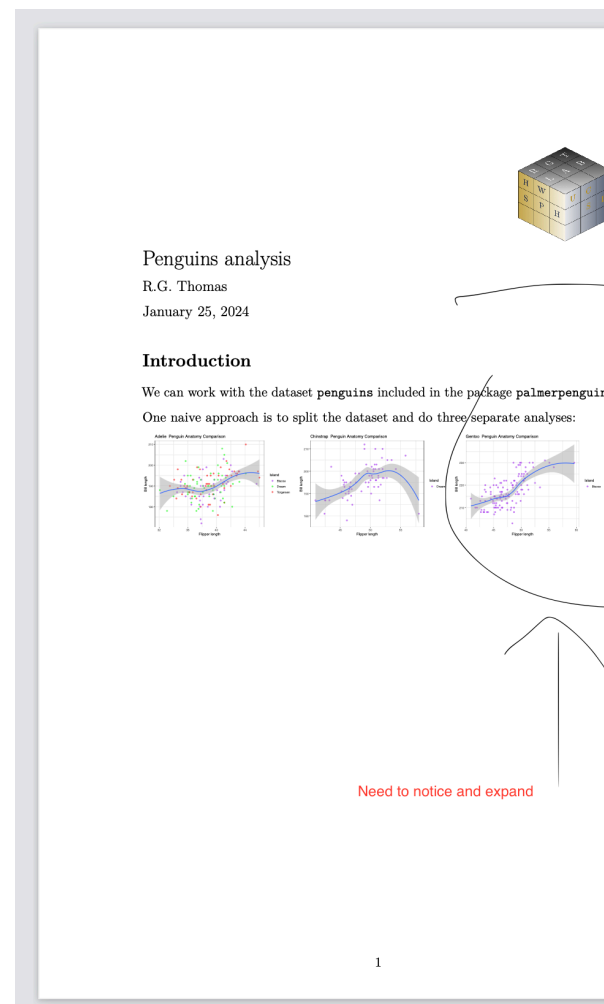


Figure 1: rendered page