

# Working example for generating multiple plots inside a map2 call

Ronald (Ryy) Glenn Thomas

2024-02-22

## Table of contents

1	Introduction	1
2	Plots for every variable and each species map inside map see ref 2 below	3
3	combine plots in a upper triangular grid with correlation coefs	3
4	Code	3
5	References	5

## 1 Introduction

Consider the problem of running a data analysis requiring a separate analysis for each of `n` strata. For example consider an effort to model the relationship between `Bill length` and `Flipper length` across three different species of penguins.

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

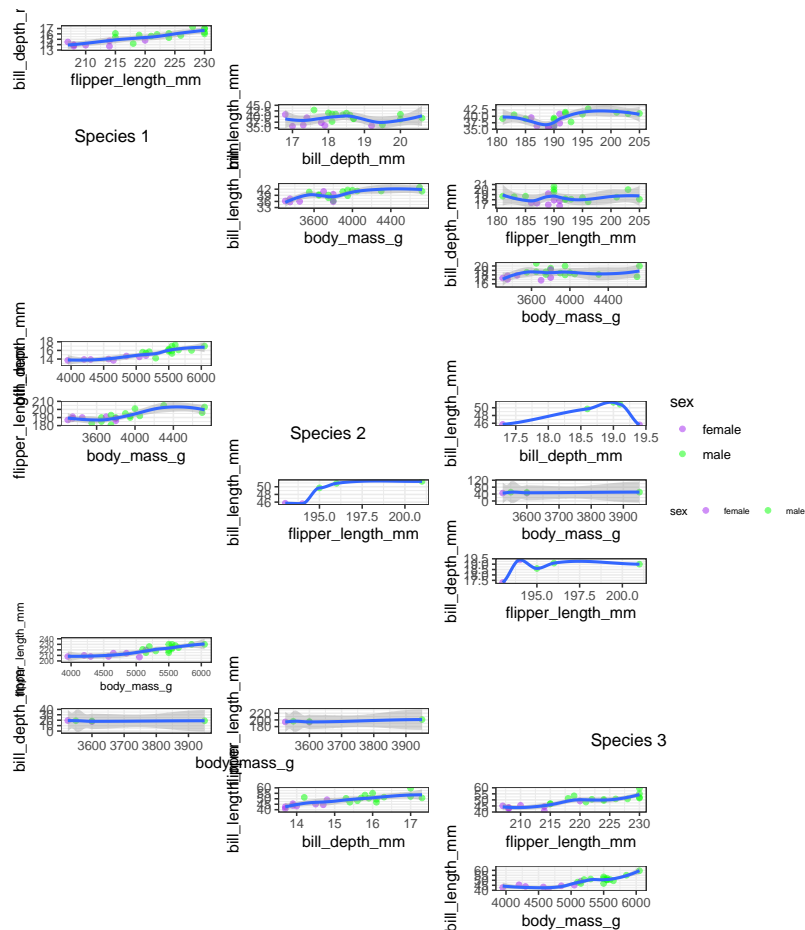


Figure 1: purrr

```
library(palmerpenguins)
```

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

The R package `purrr` provides a straightforward method to conduct the analyses with a single command. Assume the set of data tables are contained in a list of dataframes. Also assume the analysis is a simple visualization of a potential linear association between two features,



**2 Plots for every variable and each species  
map inside map see ref 2 below**

**3 combine plots in a upper triangular grid  
with correlation coefs**

## 4 Code

```
1 library(pacman)
2 p_load(grid, patchwork, rlang, purrr, palmerpenguins, tidyverse, knitr)
3
4 opts_chunk$set(
5   warning = FALSE, message = FALSE, echo = FALSE, fig.width = 8,
6   fig.height = 9, results = "asis", dev = "pdf"
7 )
8 df0 <- sample_n(penguins, 50) |> na.omit()
9 # nn = 50 ; df1 = sample_n(penguins, 50)
10 df1 <- split(df0, df0$species)
11 # df2 = penguins |> group_by(species)
12
13
14 ct <- names(df0)[3:6]
15 # mm = expand.grid(names(df1[3:6]), names(df1[3:6]))
16 nn <- t(combn(ct, 2))
17 colnames(nn) <- letters[1:2]
18 nn2 <- data.frame(nn) |> cbind(g = "sex")
19
20 zz.scatter <- function(data, formula, ...) {
21   # function to take a dataframe, and a formula with potentially a '|' group
22   #   option and return scatterplot matrix with optional R^2, loess smooth,
23   #   or least squares line.
24 }
25
26 plt1 <- function(a, b, g, spc, df_split) {
27   out_plot <- df_split |> ggplot(aes(x = .data[[a]], y = .data[[b]])) +
28     geom_point(aes(color = .data[[g]]), alpha = .5) +
29     geom_smooth() +
```

```

30     scale_color_manual(values = c("purple", "green", "red")) +
31     theme_bw()
32     assign(paste0(spc, "_", a, "_", b), value = out_plot, envir = .GlobalEnv)
33     return(out_plot)
34 }
35
36
37 temp <- df1 |> map2(names(df1), function(df_split, spc) {
38   nn2 |> pmap(function(a, b, g) {
39     plt1(b, a, g, spc, df_split)
40   })
41 })
42
43 temp2 = list_flatten(temp)
44
45 A <- temp[[1]][[1]]
46 p2 <- temp[[1]][[2]]
47 p3 <- temp[[1]][[3]]
48 p4 <- temp[[1]][[4]]
49 p5 <- temp[[1]][[5]]
50 p6 <- temp[[1]][[6]]
51 p7 <- temp[[2]][[1]]
52 p8 <- temp[[2]][[2]]
53 p9 <- temp[[2]][[3]]
54 p10 <- temp[[2]][[4]]
55 p11 <- temp[[2]][[5]]
56 p12 <- temp[[2]][[6]]
57 p13 <- temp[[3]][[1]]
58 p14 <- temp[[3]][[2]]
59 p15 <- temp[[3]][[3]]
60 p16 <- temp[[3]][[4]]
61 p17 <- temp[[3]][[5]]
62 p18 <- temp[[3]][[6]]
63 # names(temp)
64
65 layout <- "
66 X##
67 ABC
68 #DE
69 ##F

```

```

70 Y##
71 GHI
72 #JK
73 ##L
74 Z##
75 MNO
76 #PQ
77 ##R
78 "
79 X <- grid::textGrob("Species 1")
80 t2 <- grid::textGrob("Species 2")
81 t3 <- grid::textGrob("Species 3")
82
83 out <- wrap_plots(
84   X , A, B = p2, C = p3, D = p4, E = p5, F = p6, Y = t2,
85   G = p7, H = p8, I = p9, J = p10, K = p11, L = p12, Z = t3,
86   M = p13, N = p14, O = p15, P = p16, Q = p17, R = p18,
87   design = layout
88 ) +
89   plot_layout(
90     guides = "collect",
91     axis_titles = "collect"
92   ) +
93   theme(
94     legend.position = "bottom",
95     legend.direction = "horizontal",
96     text = element_text(size = 8)
97   )
98
99 out

```

## 5 References

[principal components analysis](#)

[Automating exploratory plots with ggplot2 and purrr](#)