package:cowplot

Table of Contents

update2019-08-26 17:11:39

# はじめに

パッケージ**cowplot**はggplot2の拡張である． ggplotのオブジェクトを1つのオブジェクトとして並びを作ったり， annotationを加えることが出来る

* [github](https://github.com/wilkelab/cowplot)
* [viggnette](https://cran.r-project.org/web/packages/cowplot/cowplot.pdf)

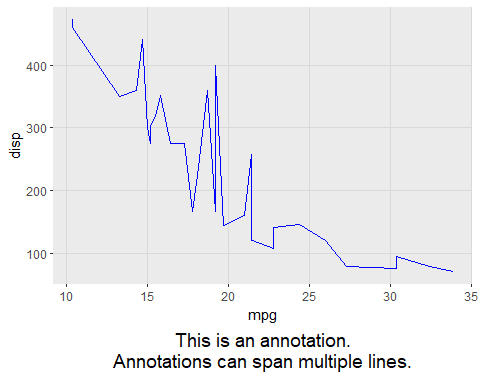
# 使用例

## add

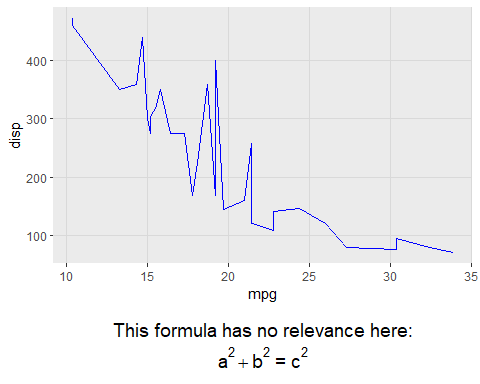
### add\_sub

グラフの下に文字列や数式を追加する関数．キャプションが必要なときになどに 使われる．

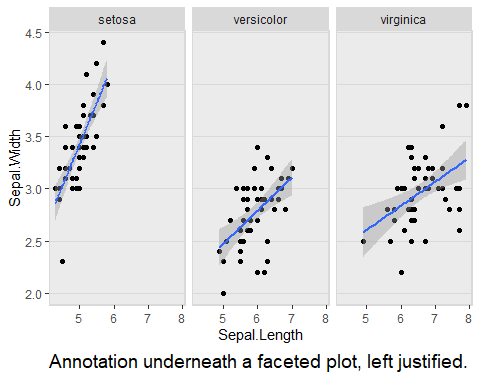
p1 <- ggplot(mtcars, aes(mpg, disp)) + geom\_line(colour = "blue") + background\_grid(minor='none')  
ggdraw(add\_sub(p1, "This is an annotation.\nAnnotations can span multiple lines."))



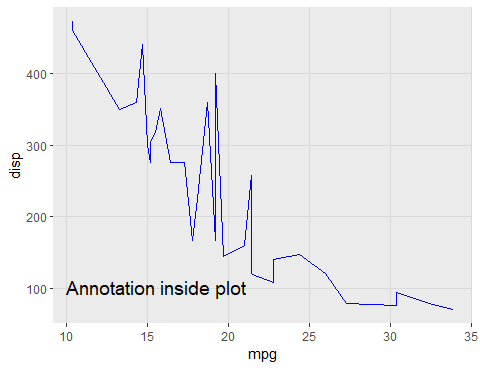
# You can also do this repeatedly.  
p2 <- add\_sub(p1, "This formula has no relevance here:", y = 0, vjust = 0)  
p3 <- add\_sub(p2, expression(paste(a^2+b^2, " = ", c^2)))  
ggdraw(p3)



#This code also works with faceted plots:  
plot.iris <- ggplot(iris, aes(Sepal.Length, Sepal.Width)) +  
geom\_point() + facet\_grid(. ~ Species) + stat\_smooth(method = "lm") +  
background\_grid(major = 'y', minor = "none") + # add thin horizontal lines  
panel\_border() # and a border around each panel  
p2 <- add\_sub(plot.iris,   
 "Annotation underneath a faceted plot, left justified.",   
 x = 0, hjust = 0)  
ggdraw(p2)



ggdraw(add\_sub(p1, "Annotation inside plot",   
 vpadding=grid::unit(0, "lines"),   
 y = 6, x = 0.03, hjust = 0))



## else

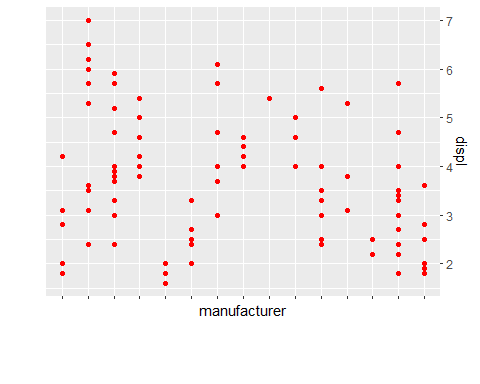
### align\_margin

align\_plots()関数のヘルパらしいがよくわかっていない.

### align\_plot

軸を調整しながらグラフを並べる技術． 例として二軸プロットが示されている． 軸をマージナルプロットのように軸を調整しながら記述することも可能なように見えるので， きちんと理解をしたい．

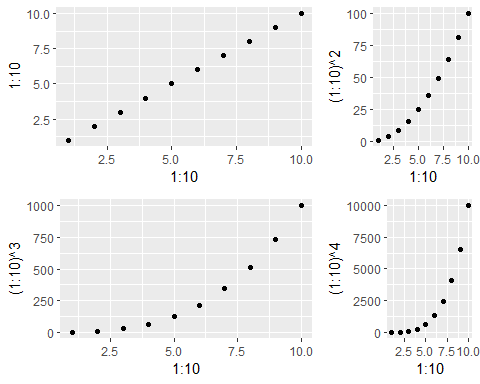
p1 <-  
 ggplot(mpg, aes(manufacturer, hwy)) +   
 stat\_summary(fun.y="median", geom = "bar") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1, vjust= 1))  
p2 <-   
 ggplot(mpg, aes(manufacturer, displ)) + geom\_point(color="red") +  
 # y軸のポジションを右側にしているのが味噌  
 # よく考えるとこれはx軸の二軸についても可能なのではないのか？  
 scale\_y\_continuous(position = "right") +  
 theme(axis.text.x = element\_blank())  
# manually align and plot on top of each other  
# alignで縦と横, あるいは両方の軸を調整するのかを確認している  
aligned\_plots <-   
 align\_plots(p1, p2, align="hv", axis="tblr")  
  
# Note: In most cases two y-axes should not be used, but this example  
# illustrates how one would could accomplish it.  
ggdraw(aligned\_plots[[1]]) + draw\_plot(aligned\_plots[[2]])



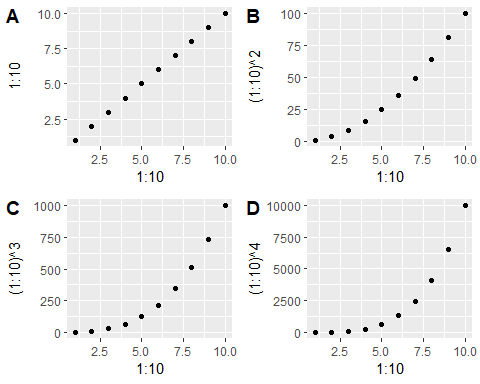
### plot\_grid

グリッド配置をしながら軸についても調整を行う.

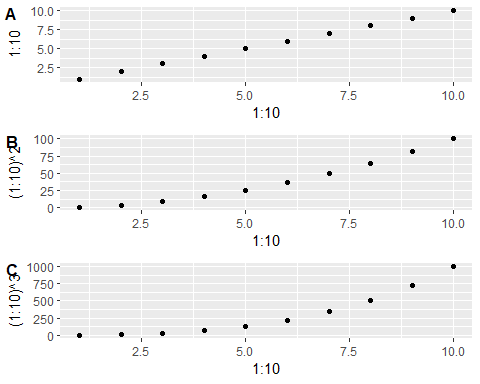
p1 <- qplot(1:10, 1:10)  
p2 <- qplot(1:10, (1:10)^2)  
p3 <- qplot(1:10, (1:10)^3)  
p4 <- qplot(1:10, (1:10)^4)  
p5 <- ggplot(mpg, aes(as.factor(year), hwy)) +  
geom\_boxplot() +  
facet\_wrap(~class, scales = "free\_y")  
# simple grid  
plot\_grid(p1, p2, p3, p4, rel\_widths = c(2,1))



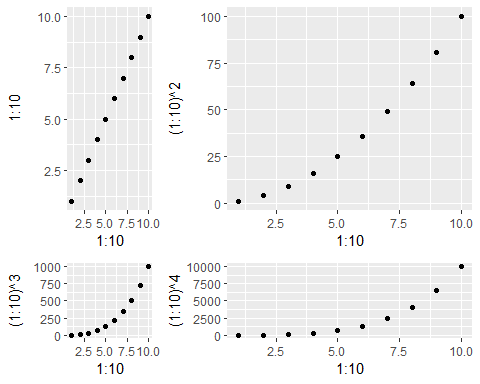
# simple grid with labels and aligned plots  
plot\_grid(p1, p2, p3, p4, labels=c('A', 'B', 'C', 'D'), align="hv")



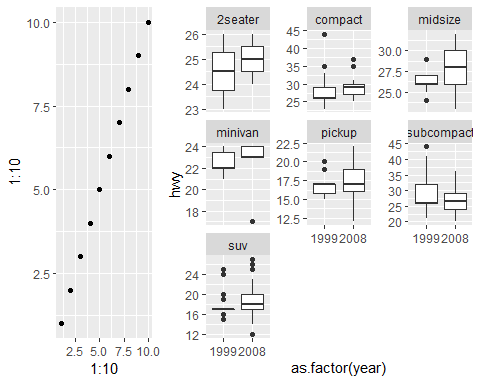
# manually setting the number of rows, auto-generate upper-case labels  
plot\_grid(p1, p2, p3, nrow=3, labels="AUTO", label\_size=12, align="v")



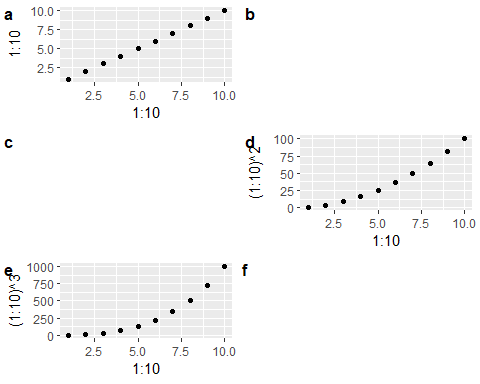
# making rows and columns of different widths/heights  
plot\_grid(p1, p2, p3, p4, align='hv', rel\_heights=c(2,1), rel\_widths=c(1,2))



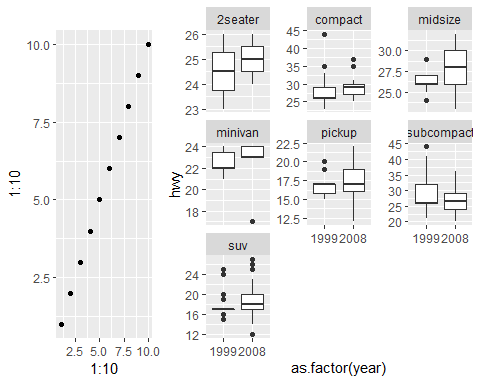
# aligning complex plots in a grid  
plot\_grid(p1, p5, align="h", axis="b", nrow = 1, rel\_widths = c(1,2))



# more examples  
#' # missing plots in some grid locations, auto-generate lower-case labels  
plot\_grid(p1, NULL, NULL, p2, p3, NULL, ncol=2,  
labels="auto", label\_size=12, align="v")



# can align top of plotting area as well as bottom  
plot\_grid(p1, p5, align="h", axis="tb", nrow = 1, rel\_widths = c(1,2))



# other types of plots not generated with ggplot  
dev.new()  
par(xpd = NA, # switch off clipping, necessary to always see axis labels  
bg = "transparent", # switch off background to avoid obscuring adjacent plots  
oma = c(2, 2, 0, 0), # move plot to the right and up  
mgp = c(2, 1, 0) # move axis labels closer to axis  
)  
plot(sqrt)  
p6 <- recordPlot()  
dev.off()

## png   
## 2

p7 <- function() image(volcano)  
p8 <- grid::circleGrob()  
## 次のコマンドはguiで動かす際にはエラーにはならないけれど，markdownではエラーになる  
# plot\_grid(p1, p6, p7, p8, labels = "AUTO", scale = c(1, 1, .85, .9))

これはとても便利であると思うので，もう少し情報を調べる必要がある． いやーしかし，便利ですわ．

### marginal プロット

ということは、周辺プロットも作成が出来るはず.

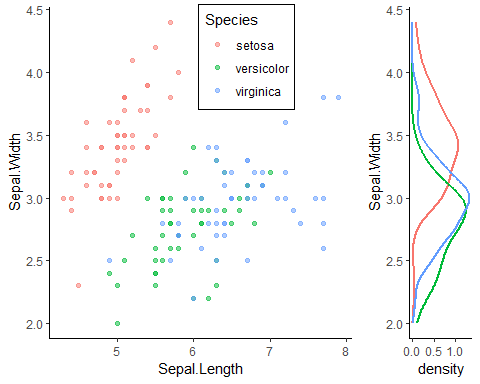
p1 <-   
 iris %>%   
 ggplot(aes(Sepal.Length, Sepal.Width, color = Species)) +   
 geom\_point(size = 1.5, alpha = .5) +   
 theme\_classic() +   
 theme(legend.position = c(0.65,0.85),   
 legend.background = element\_rect(color = "black"))   
  
p2 <-   
 iris %>%  
 ggplot(aes(Sepal.Width, color = Species)) +   
 geom\_line(stat = "density", size = 1, shape = 1) +   
 coord\_flip() +   
 theme\_classic() +   
 guides(color = "none")

## Warning: Ignoring unknown parameters: shape

scale\_x\_continuous(position = "top")

## <ScaleContinuousPosition>  
## Range:   
## Limits: 0 -- 1

plot\_grid(p1, p2, nrow = 1, rel\_widths = c(3,1))



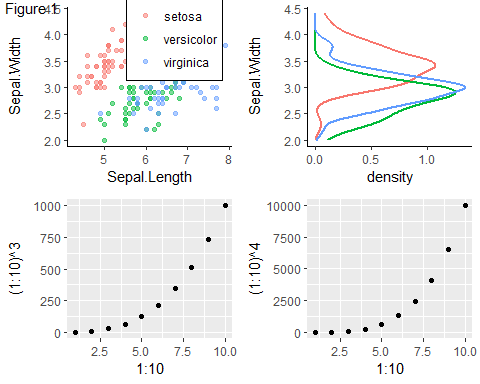
## draw

**draw\_?**系の関数はanotationを行える．これにより，プロット領域にテキストを 書いたりすることが楽に行える．

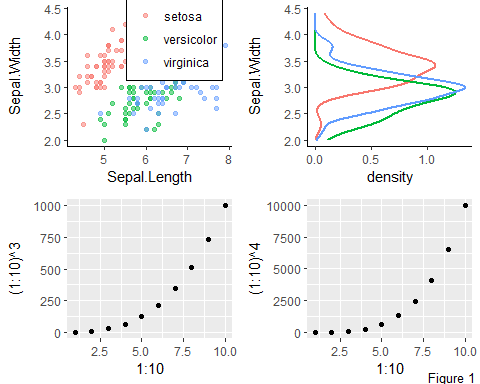
* draw\_figure\_label
* draw\_grob
* draw\_image
* draw\_label
* draw\_line
* draw\_plot
* draw\_plot\_label
* draw\_text

### draw\_figure\_label

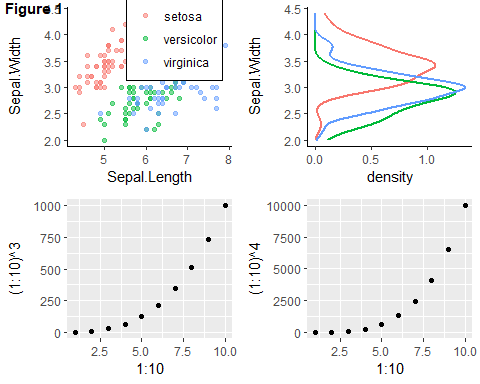
# Create a simple grid  
p <- plot\_grid(p1, p2, p3, p4, align = 'hv')  
  
# Default font size and position  
p + draw\_figure\_label(label = "Figure 1")



# Different position and font size  
p + draw\_figure\_label(label = "Figure 1", position = "bottom.right", size = 10)

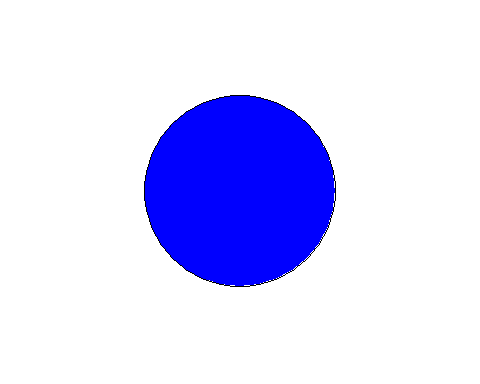


# Using bold font face  
p + draw\_figure\_label(label = "Figure 1", fontface = "bold")



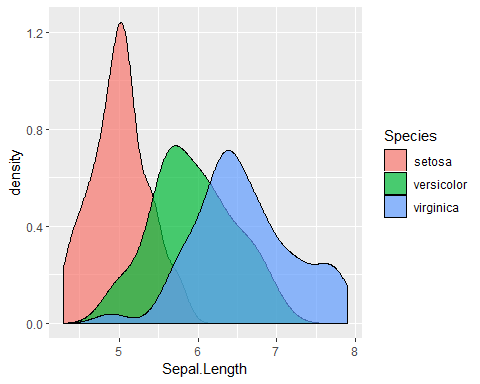
### draw\_grob

# A grid grob (here a blue circle)  
library(grid)  
g <- circleGrob(gp = gpar(fill = "blue"))  
# place into the middle of the plotting area, at a scale of 50%  
ggdraw() + draw\_grob(g, scale = 0.5)

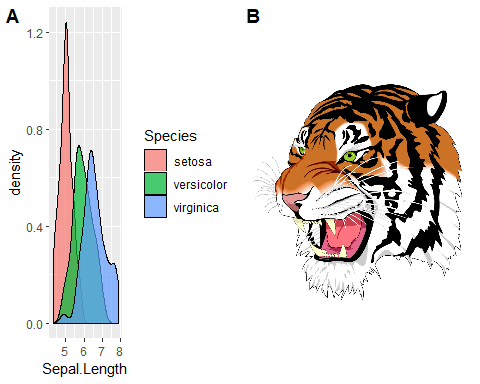


### draw\_image

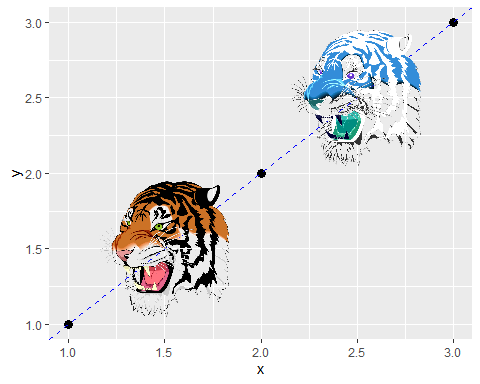
# Use image as plot background  
p <- ggplot(iris, aes(x = Sepal.Length, fill = Species)) + geom\_density(alpha = 0.7)  
ggdraw() +  
 draw\_image("http://jeroen.github.io/images/tiger.svg") +  
 draw\_plot(p + theme(legend.box.background = element\_rect(color = "white")))



# Make grid with plot and image  
p <- ggplot(iris, aes(x = Sepal.Length, fill = Species)) +  
 geom\_density(alpha = 0.7)  
p2 <- ggdraw() + draw\_image("http://jeroen.github.io/images/tiger.svg", scale = 0.9)  
plot\_grid(p, p2, labels = "AUTO")



# Manipulate images and draw in plot coordinates  
if (requireNamespace("magick", quietly = TRUE)){  
 img <- magick::image\_read("http://jeroen.github.io/images/tiger.svg")  
 img <- magick::image\_transparent(img, color = "white")  
 img2 <- magick::image\_negate(img)  
 ggplot(data.frame(x = 1:3, y = 1:3), aes(x, y)) +  
 geom\_point(size = 3) +  
 geom\_abline(slope = 1, intercept = 0, linetype = 2, color = "blue") +  
 draw\_image(img , x = 1, y = 1, scale = .9) +  
 draw\_image(img2, x = 2, y = 2, scale = .9)  
}



### draw\_line

ggdraw() + draw\_line(x = c(0.2, 0.7, 0.7, 0.3),  
 y = c(0.1, 0.3, 0.9, 0.8),  
 color = "blue", size = 2)

