package:cowplot

Table of Contents

update2019-08-21 10:37:59

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
- viggnette

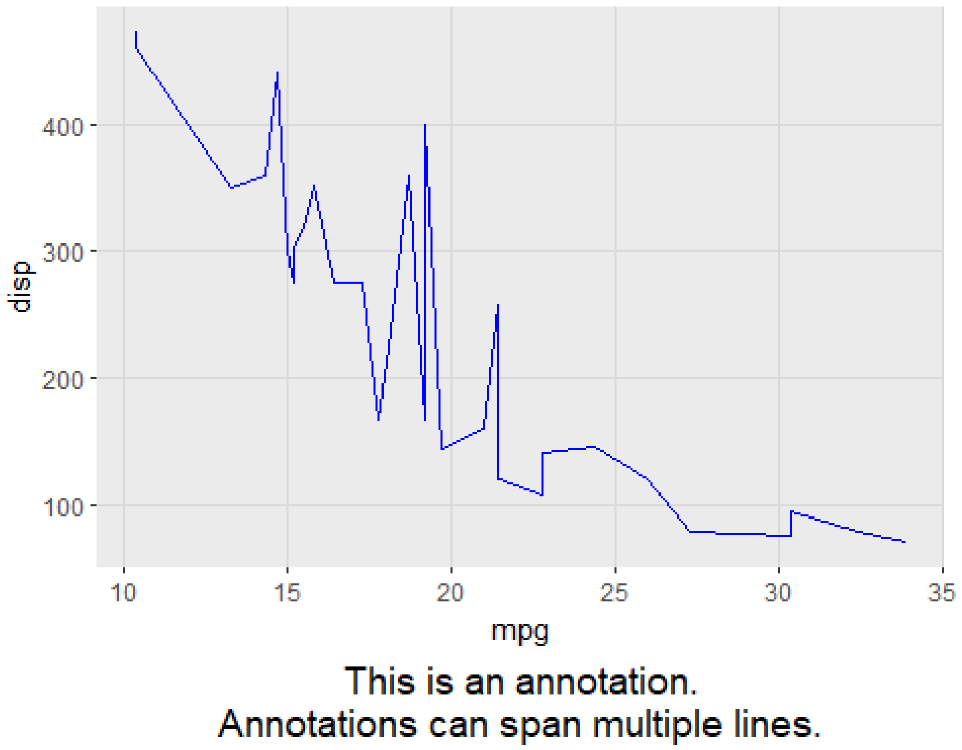
使用例

add

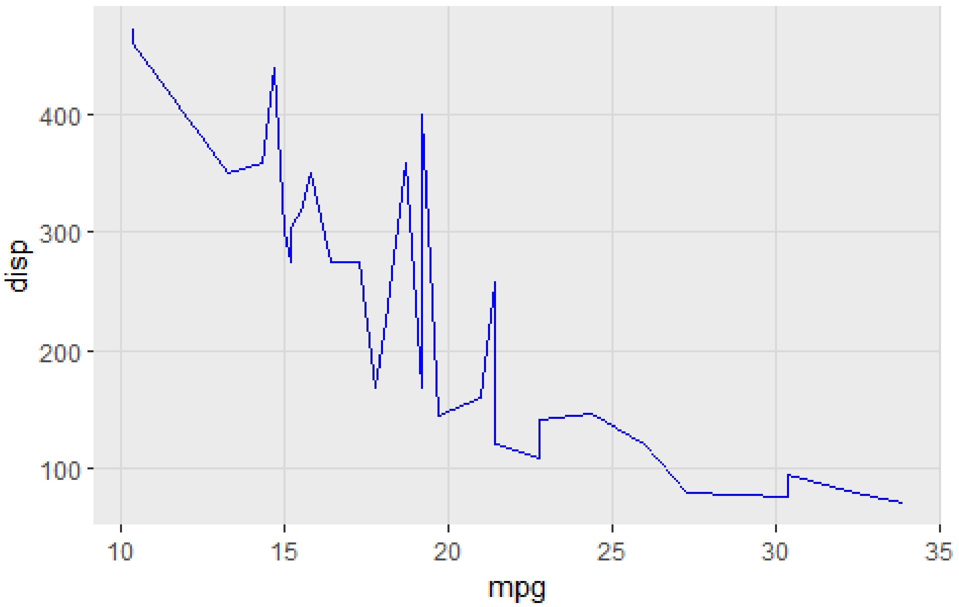
add_sub

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") + backg
round_grid(minor='none')
ggdraw(add_sub(p1, "This is an annotation.\nAnnotations can span multiple
 lines."))
```



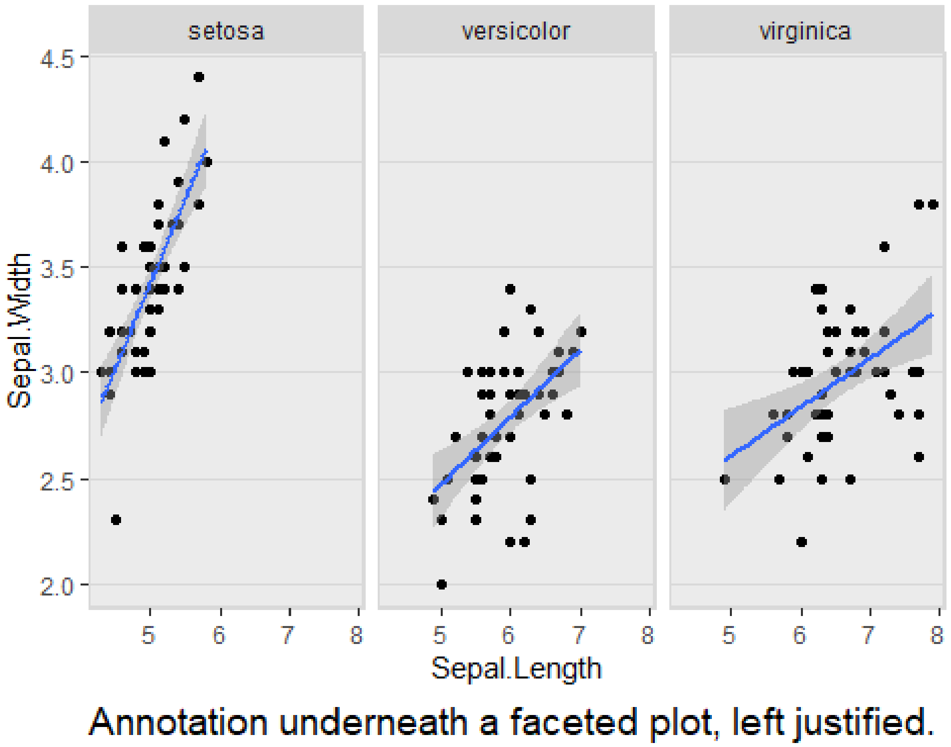
```
# You can also do this repeatedly.
p2 <- add_sub(p1, "This formula has no relevance here:", y = 0, vjust =
0)
p3 <- add_sub(p2, expression(paste(a^2+b^2, " = ", c^2)))
ggdraw(p3)
```



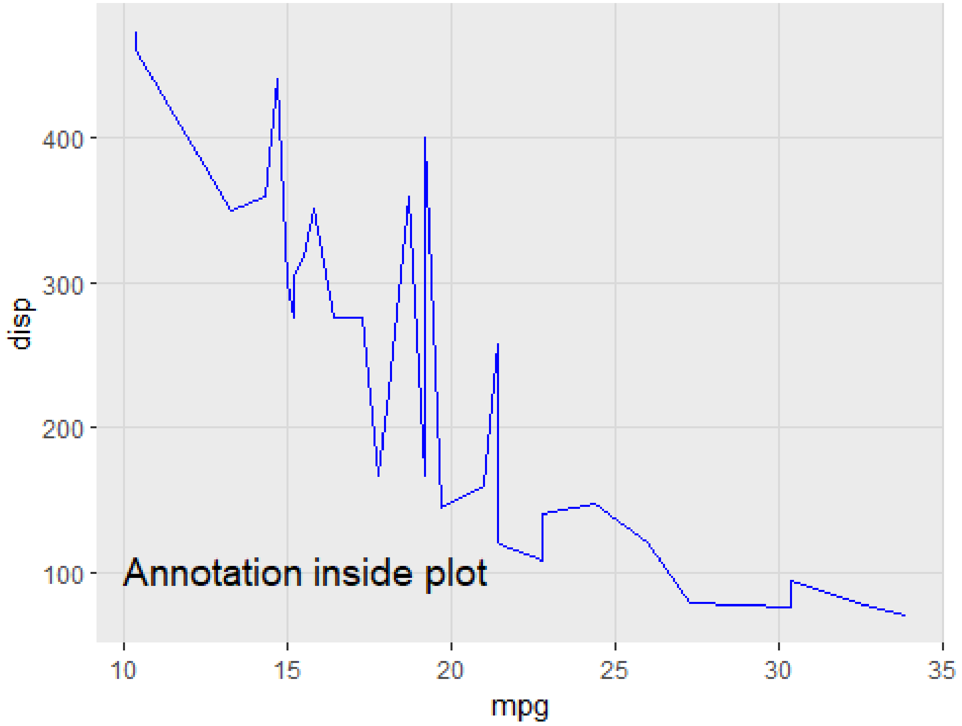
This formula has no relevance here:

$$a^2 + b^2 = c^2$$

```
#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 line
s
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

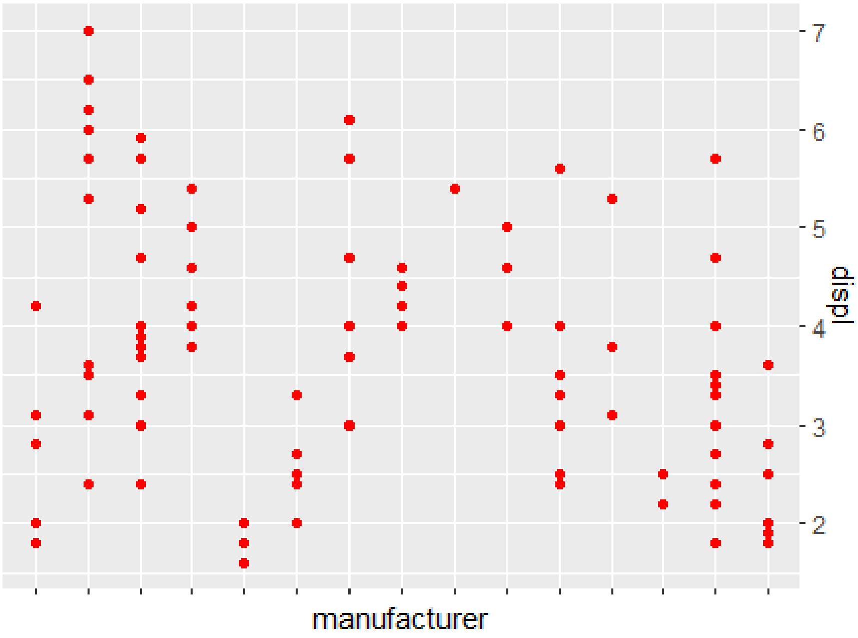
align_plots()関数のヘルパらしいがよくわかっていない.

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())
```

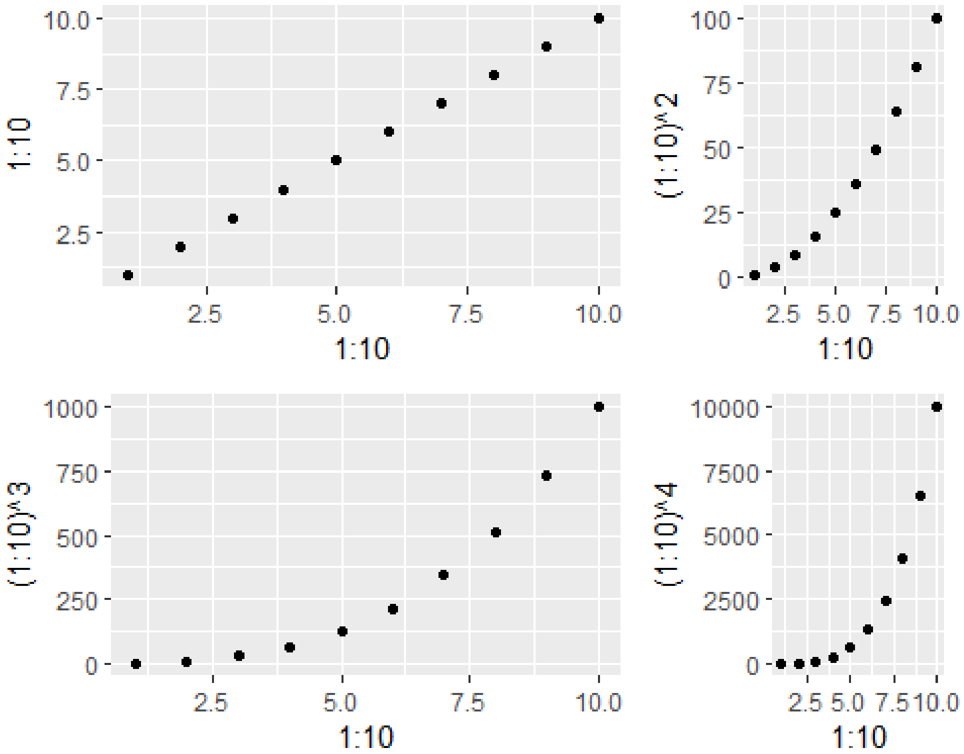
```
# manually align and plot on top of each other
# align で縦と横，あるいは両方の軸を調整するのかを確認している
aligned_plots <-
    align_plots(p1, p2, align="hv", axis="tblr")

# 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]])
```



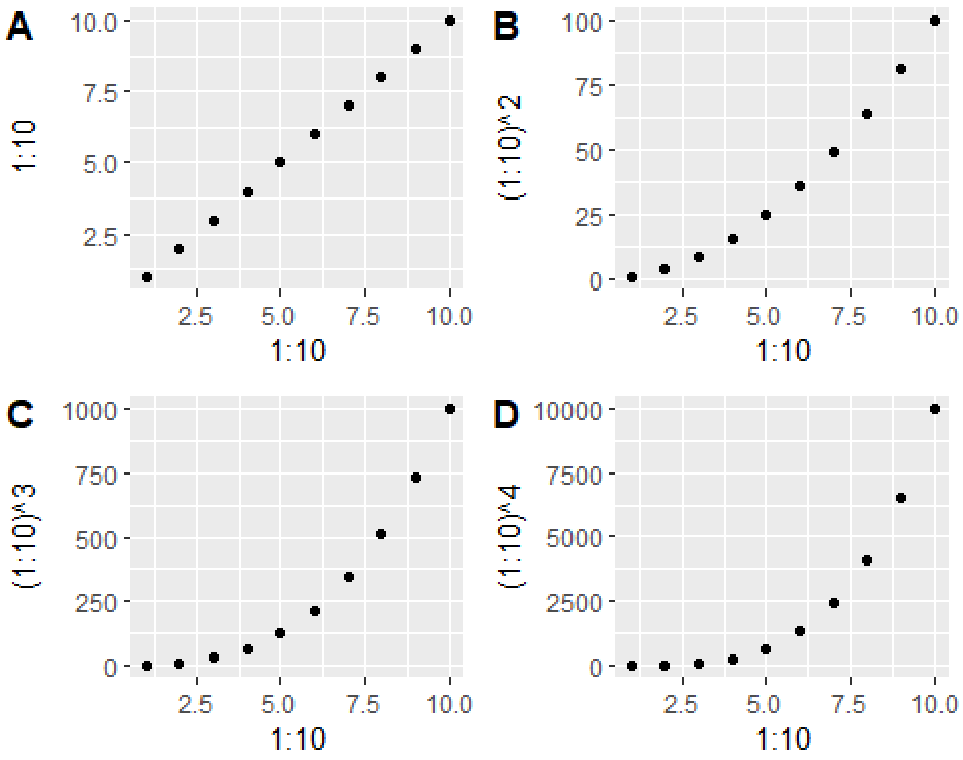
plot_grid

グリッド配置をしながら軸についても調整を行う．

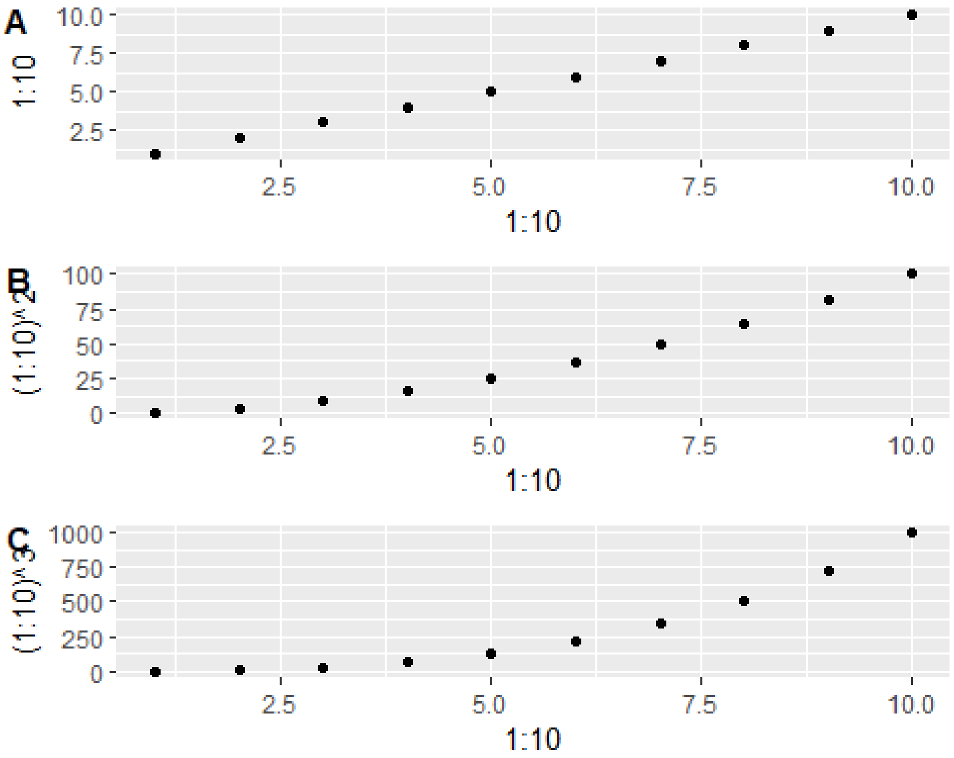
```
p1 <- qplot(1:10, 1:10)
p2 <- qplot(1:10, (1:10)^2)
p3 <- qplot(1:10, (1:10)^3)
p4 <- qplot(1:10, (1:10)^4)
p5 <- ggplot(mpg, aes(as.factor(year), hwy)) +
geom_boxplot() +
facet_wrap(~class, scales = "free_y")
# simple grid
plot_grid(p1, p2, p3, p4, rel_widths = c(2,1))
```



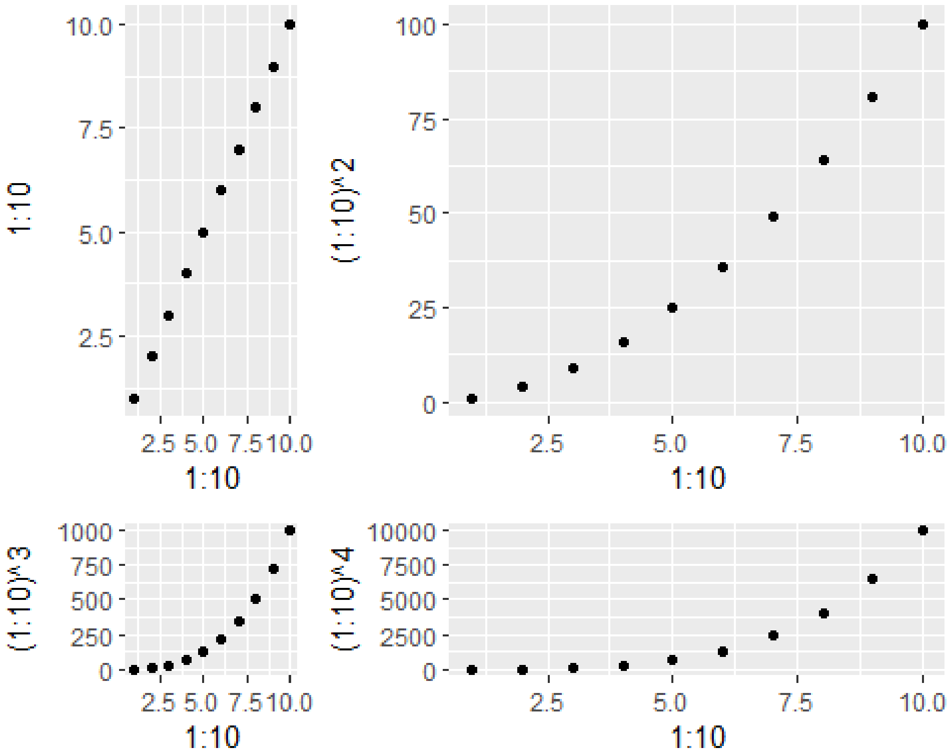
```
# simple grid with labels and aligned plots
plot_grid(p1, p2, p3, p4, labels=c('A', 'B', 'C', 'D'), align="hv")
```



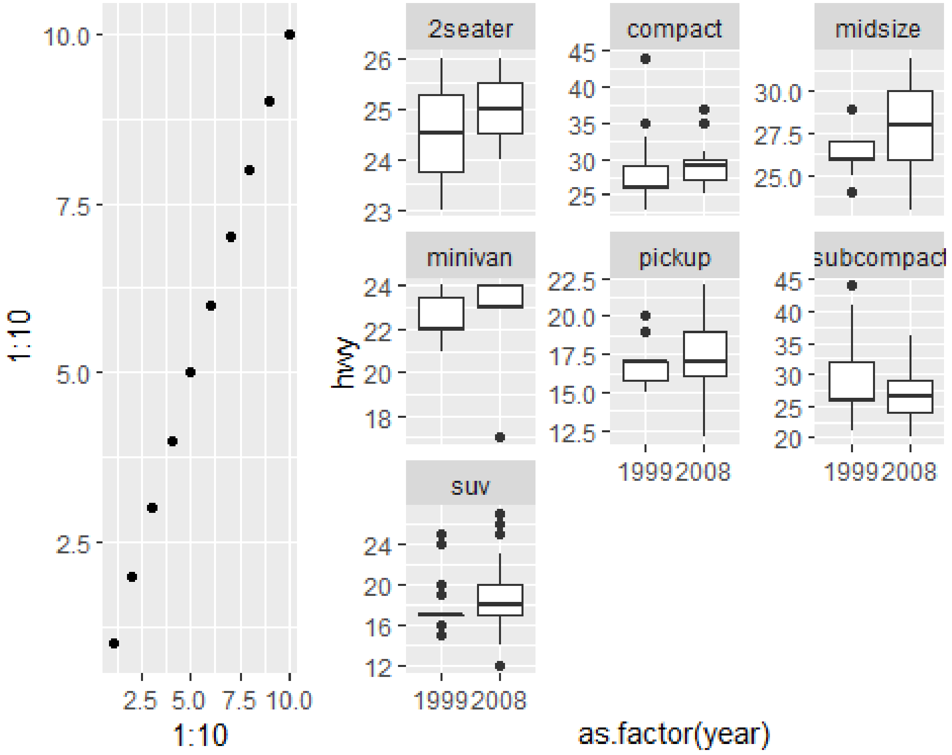
```r
# manually setting the number of rows, auto-generate upper-case labels
plot_grid(p1, p2, p3, nrow=3, labels="AUTO", label_size=12, align="v")
```



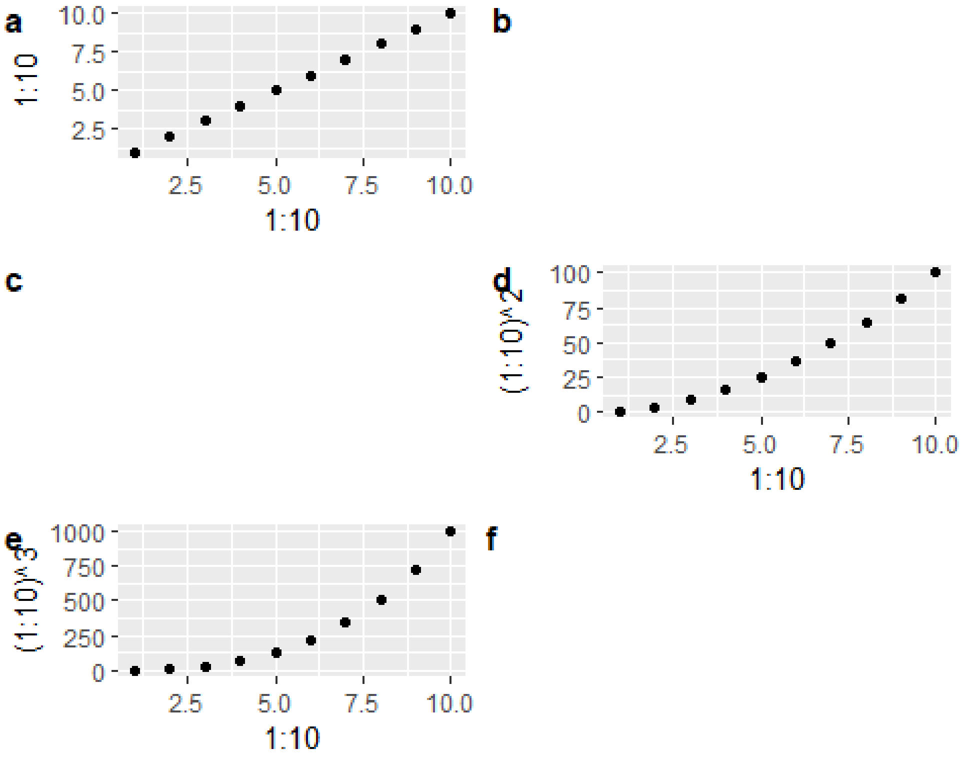
```
# making rows and columns of different widths/heights
plot_grid(p1, p2, p3, p4, align='hv', rel_heights=c(2,1), rel_widths=c(1,
2))
```



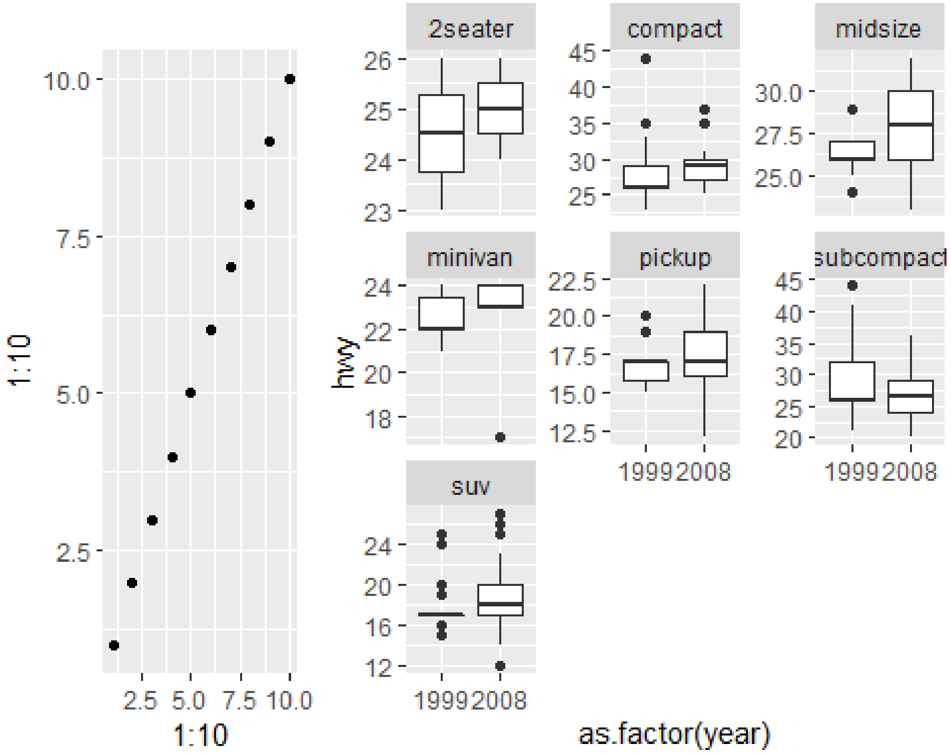
```
# aligning complex plots in a grid
plot_grid(p1, p5, align="h", axis="b", nrow = 1, rel_widths = c(1,2))
```



```
# more examples
#' # missing plots in some grid locations, auto-generate lower-case label
s
plot_grid(p1, NULL, NULL, p2, p3, NULL, ncol=2,
labels="auto", label_size=12, align="v")
```



```r
# 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))
```



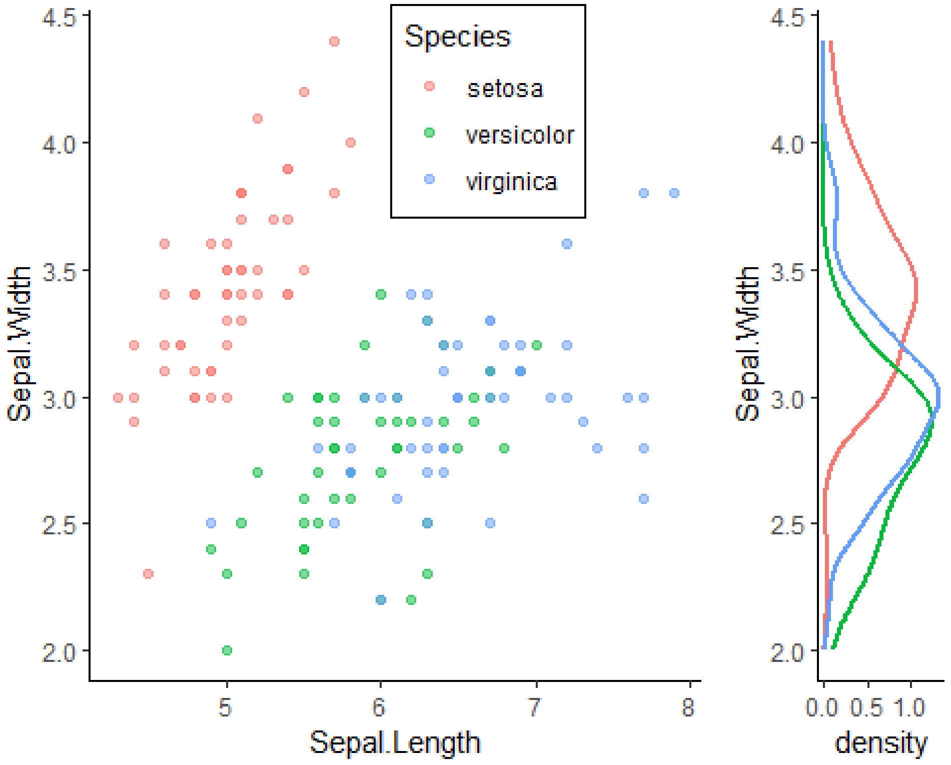
```
# other types of plots not generated with ggplot
dev.new()
par(xpd = NA, # switch off clipping, necessary to always see axis labels
bg = "transparent", # switch off background to avoid obscuring adjacent p
lots
oma = c(2, 2, 0, 0), # move plot to the right and up
mgp = c(2, 1, 0) # move axis labels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



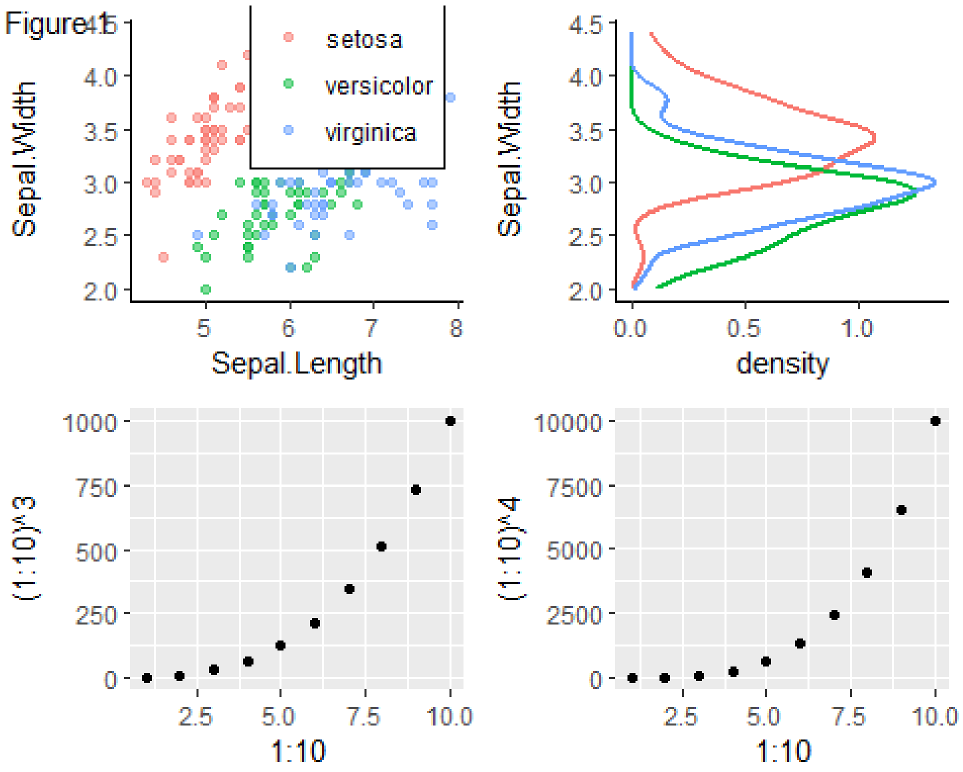
draw

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．

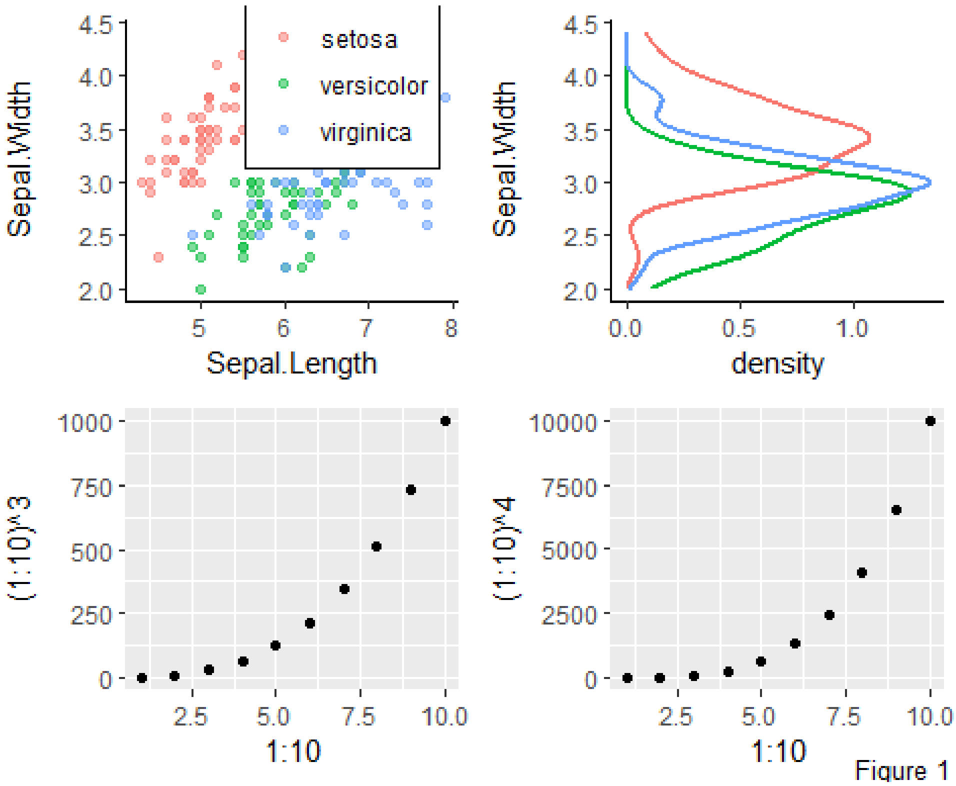
- draw_figure_label
- draw_grob
- draw_image
- draw_label
- draw_line
- draw_plot
- draw_plot_label
- draw_text

draw_figure_label

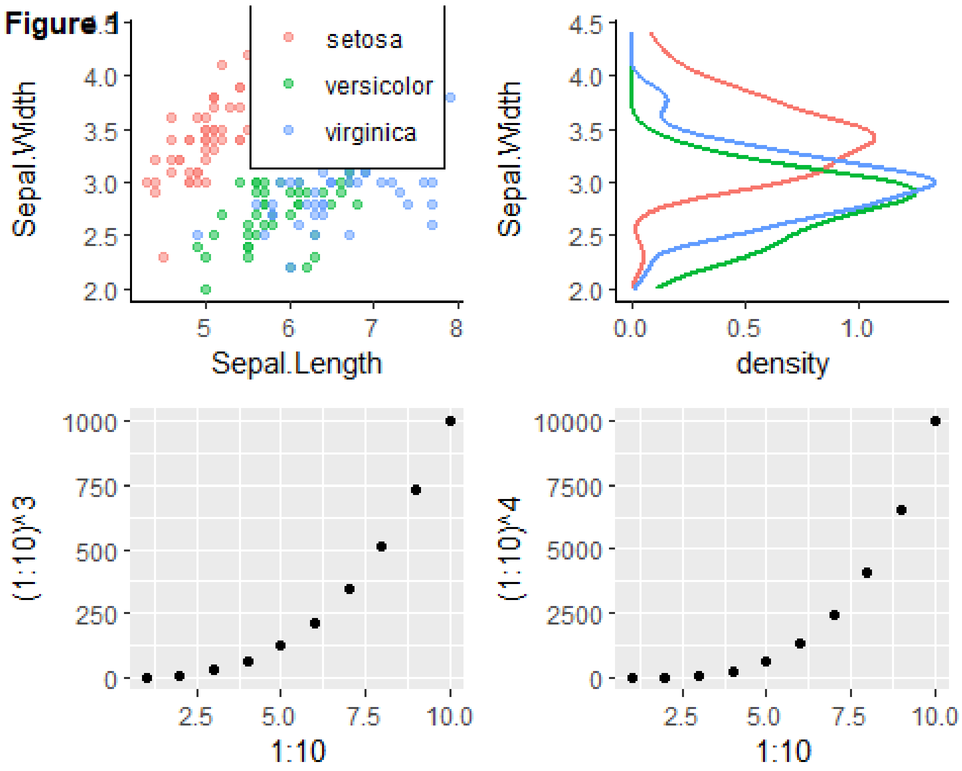
```
# Create a simple grid
p <- plot_grid(p1, p2, p3, p4, align = 'hv')

# 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)
```

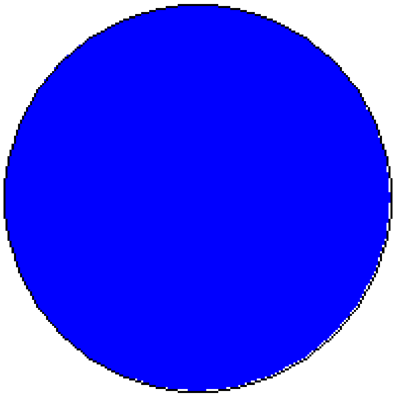


```r
# Using bold font face
p + draw_figure_label(label = "Figure 1", fontface = "bold")
```



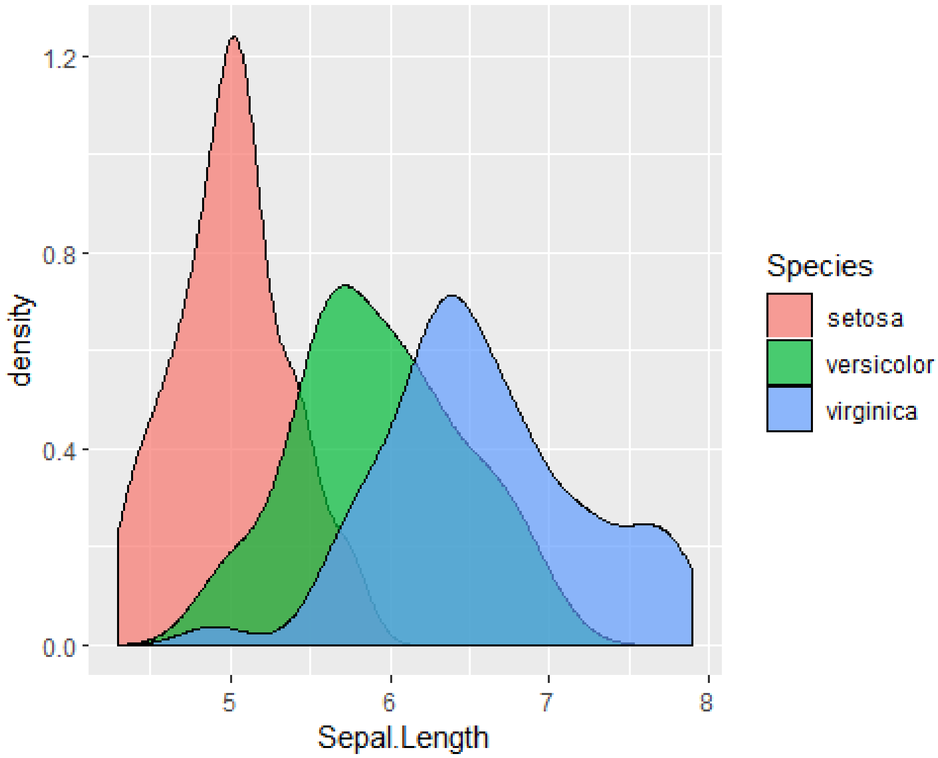
draw_grob

```
# 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)
```

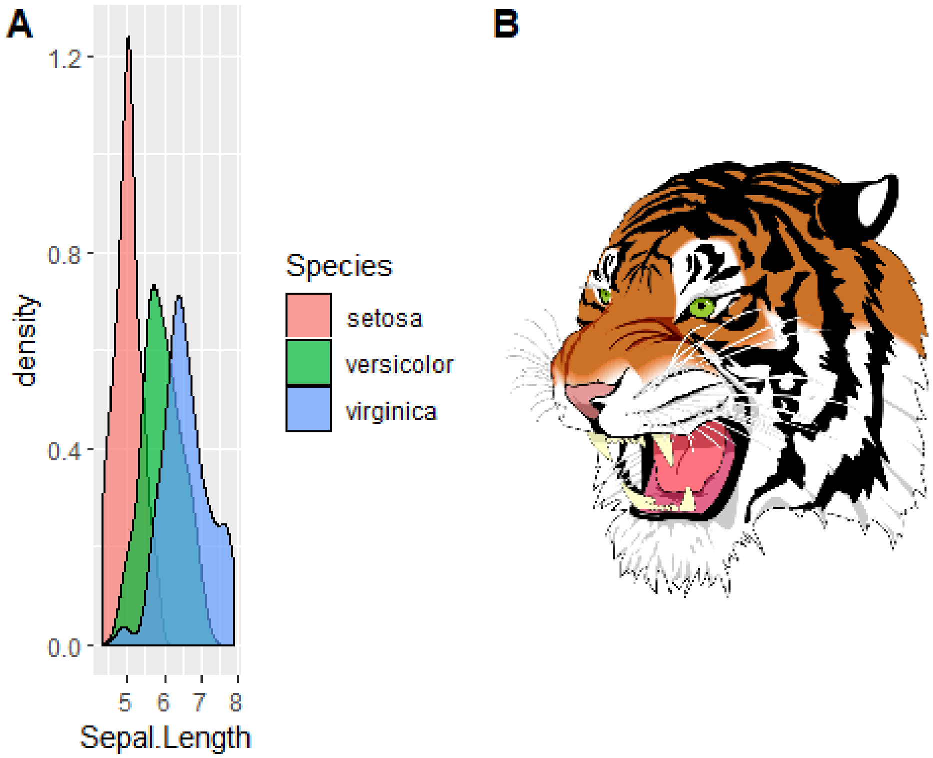


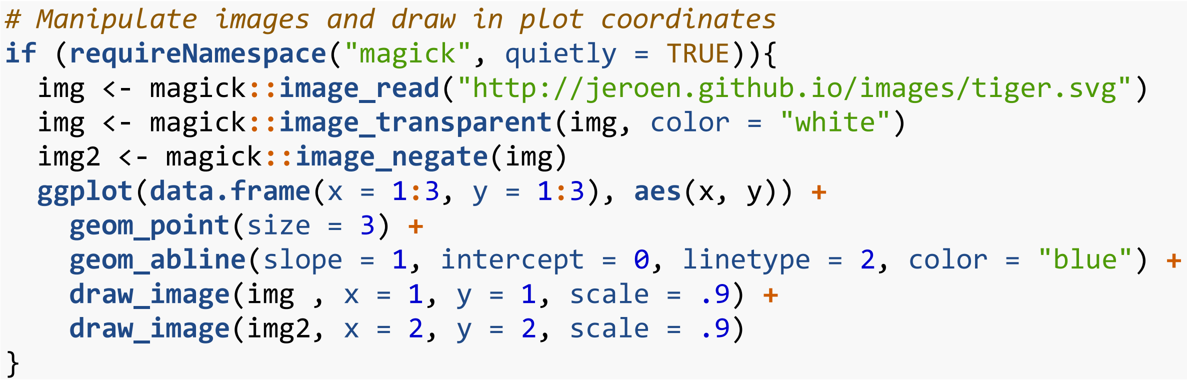
draw_image

```
# 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")))
```

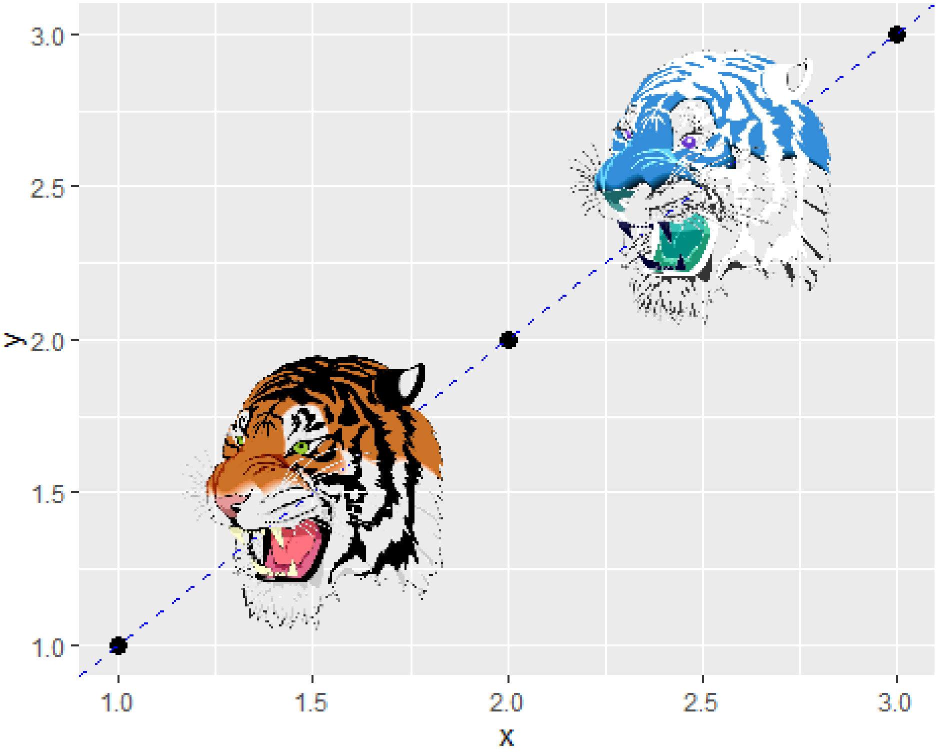


```
# Make grid with plot and image
p <- ggplot(iris, aes(x = Sepal.Length, fill = Species)) +
  geom_density(alpha = 0.7)
p2 <- ggdraw() + draw_image("http://jeroen.github.io/images/tiger.svg", s
cale = 0.9)
plot_grid(p, p2, labels = "AUTO")
```





```r
# 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)
```