Package in R: Cowplot

2023-04-26

Project 1

Data: April 26, 2023 Subject: Package in R: Cowplot Team: Gail Goveas, Abhimanyu Gupta, Gahyeon Back

```
#Load the data
data <- read.csv("World Happiness Report.csv")

#Display the first 10th raw data
head(data, 10)
```

	Country.Name <chr>	Regional.Indicator <chr>	Year <int>	Life.Ladder <dbl>	Log.GDP.Per.Capita <dbl>	Social.Support <dbl>
1	Afghanistan	South Asia	2008	3.723590	7.350416	0.4506623
2	Afghanistan	South Asia	2009	4.401778	7.508646	0.5523084
3	Afghanistan	South Asia	2010	4.758381	7.613900	0.5390752
4	Afghanistan	South Asia	2011	3.831719	7.581259	0.5211036
5	Afghanistan	South Asia	2012	3.782938	7.660506	0.5206367
6	Afghanistan	South Asia	2013	3.572100	7.680333	0.4835519
7	Afghanistan	South Asia	2014	3.130896	7.670638	0.5255684
8	Afghanistan	South Asia	2015	3.982855	7.653833	0.5285972
9	Afghanistan	South Asia	2016	4.220169	7.650370	0.5590718
10	Afghanistan	South Asia	2017	2.661718	7.647830	0.4908801

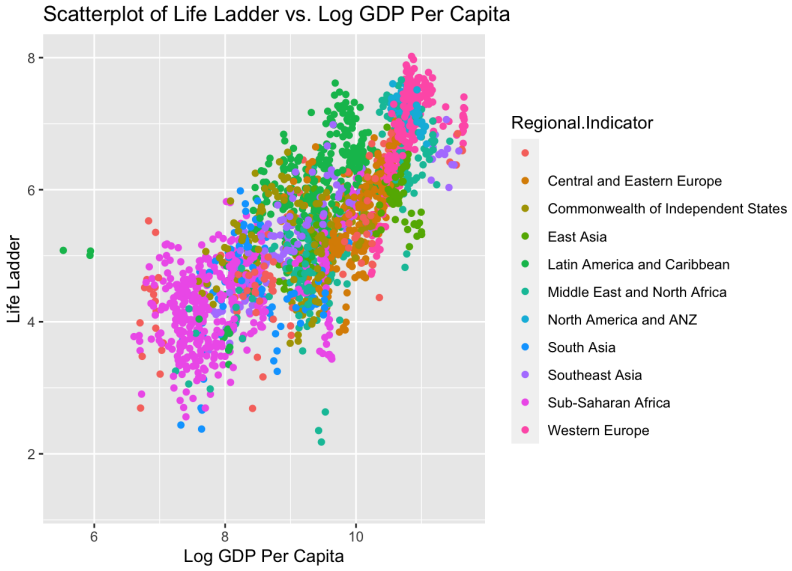
1-10 of 10 rows | 1-7 of 14 columns

```
#Load Library
library(ggplot2)
library(cowplot)
```

Case 1) Combine two plots into a single grid

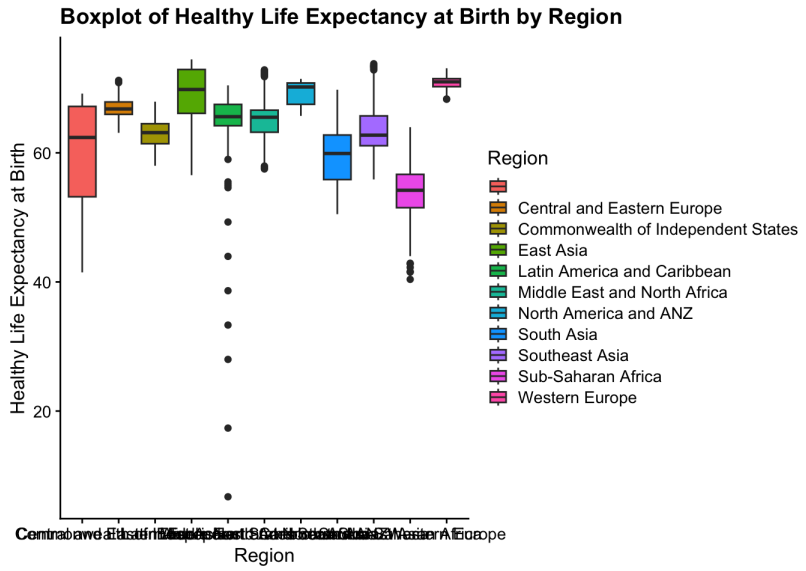
```
# Prepare data
df <- na.omit(data)
df <- data[is.finite(data$`Healthy.Life.Expectancy.At.Birth`),]

# Plot 1: Scatterplot between Life Ladder and Log GDP Per Capita
plot1 <- ggplot(data, aes(x = `Log.GDP.Per.Capita`, y = `Life.Ladder`, color = `Regional.Indicator`)) +
  geom_point() +
  labs(x = "Log GDP Per Capita", y = "Life Ladder") +
  labs(title = "Scatterplot of Life Ladder vs. Log GDP Per Capita")
print(plot1)
```

```
## Warning: Removed 20 rows containing missing values (`geom_point()`).
```



```
# Plot 2: Boxplot of Healthy Life Expectancy at Birth by Region
plot2 <- ggplot(data, aes(x = `Regional.Indicator`, y = `Healthy.Life.Expectancy.At.Birth`, fill = `Regional.Indicator`)) +
  geom_boxplot() +
  labs(x = "Region", y = "Healthy Life Expectancy at Birth", fill = "Region") +
  theme_cowplot(12) +
  labs(title = "Boxplot of Healthy Life Expectancy at Birth by Region")
print(plot2)
```

```
## Warning: Removed 54 rows containing non-finite values (`stat_boxplot()`).
```



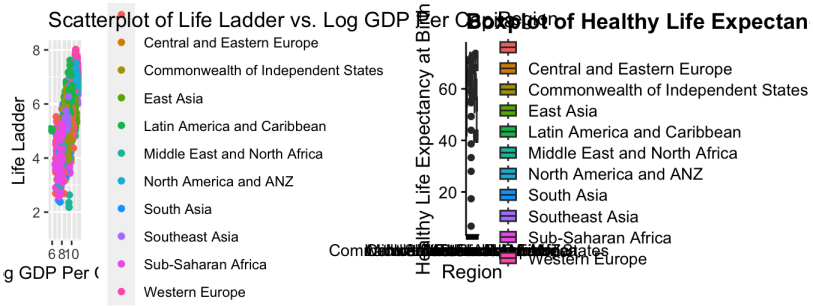

```
# Combine the two plots using cowplot
combined_plot <- plot_grid(plot1, plot2, ncol = 2, align = "v", axis = "tb", widths = c(3, 2)) +
  labs(title = "Combined Plot")
```

```
## Warning: Removed 20 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 54 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning in as_grob.default(plot): Cannot convert object of class numeric into a
## grob.
```

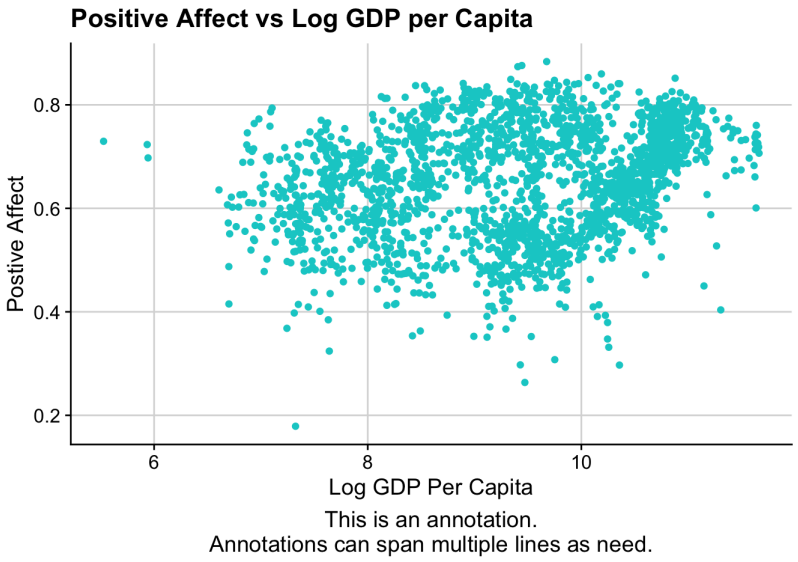
```
# Set the size of the combined plot
theme_set(theme_cowplot())
options(repr.plot.width = 10, repr.plot.height = 5)

# Print the combined plot
print(combined_plot)
```



Case 2) Adding annotations to a plot

```
p1 <- ggplot(df, aes(Log.GDP.Per.Capita, Positive.Affect)) + geom_point(colour = "cyan3") +
  background_grid(minor='none')+
  labs(x = "Log GDP Per Capita", y = "Postive Affect") +
  labs(title = "Positive Affect vs Log GDP per Capita")
ggdraw(add_sub(p1, "This is an annotation.\nAnnotations can span multiple lines as need."))
```

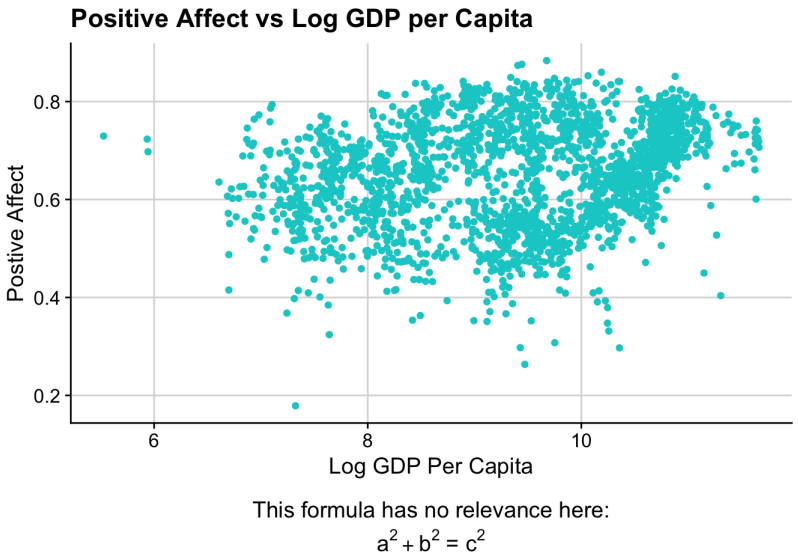
```
## Warning: Removed 35 rows containing missing values (`geom_point()`).
```




```
# You can also do this repeatedly.
p2 <- add_sub(p1, "This formula has no relevance here:", y = 0, vjust = 0)
```

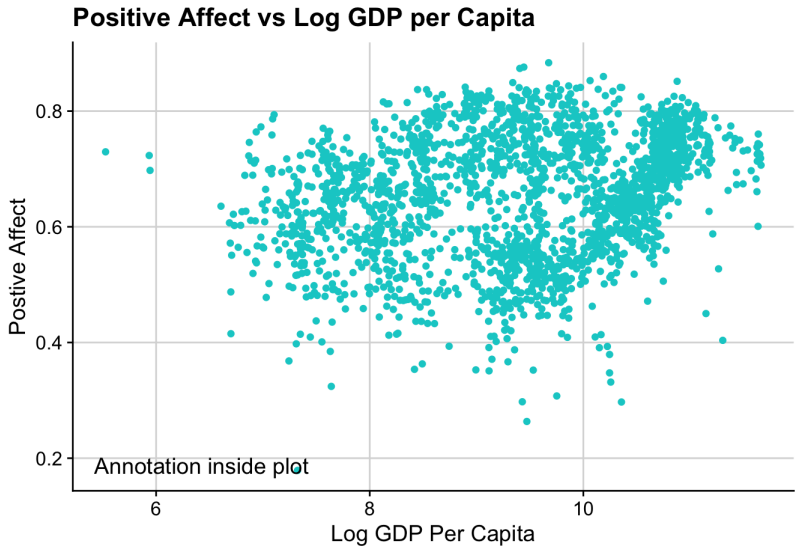
```
## Warning: Removed 35 rows containing missing values (`geom_point()`).
```

```
p3 <- add_sub(p2, expression(paste(a^2+b^2, " = ", c^2)))
ggdraw(p3)
```




```
# Finally, it is possible to move the annotation inside of the plot if desired.
ggdraw(add_sub(p1, "Annotation inside plot", vpadding=grid::unit(0, "lines"),y = 6, x = 0.03, hjust = 0))
```

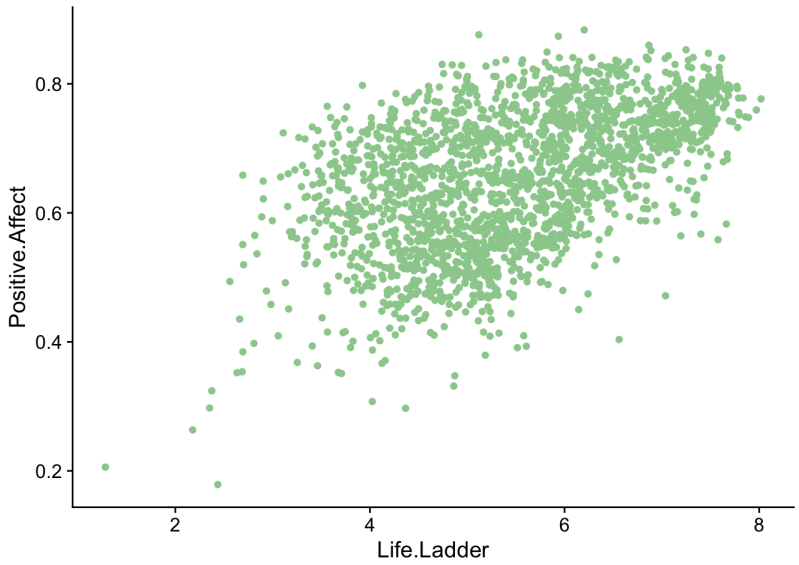
```
## Warning: Removed 35 rows containing missing values (`geom_point()`).
```



Case 3) Adding the background grid in a ggplot2 plot

```
# the plot without a background grid
ggplot(df, aes(Life.Ladder, Positive.Affect)) +
  geom_point(colour = "darkseagreen3")
```

```
## Warning: Removed 24 rows containing missing values (`geom_point()`).
```




```
#using cowplot to add a background grid
ggplot(df, aes(Life.Ladder, Positive.Affect)) +
  geom_point(colour = "darkseagreen3") +
  theme_half_open() +
  background_grid()
```

```
## Warning: Removed 24 rows containing missing values (`geom_point()`).
```

Case 4) Adding label to the figure

```
p_1 <- ggplot(df, aes(Life.Ladder, Positive.Affect)) +
  geom_point(colour = "pink1")


p_2 <- ggplot(df, aes(Log.GDP.Per.Capita, Positive.Affect)) +
  geom_point(colour = "lightblue")


p_3 <- ggplot(df, aes(Social.Support, Positive.Affect)) +
  geom_point(colour = "thistle2")

p_4 <- ggplot(df, aes(Freedom.To.Make.Life.Choices, Positive.Affect)) +
  geom_point(colour = "lightcoral")

# Create a simple grid
p <- plot_grid(p_1, p_2,p_3, p_4, align = 'hv')
```

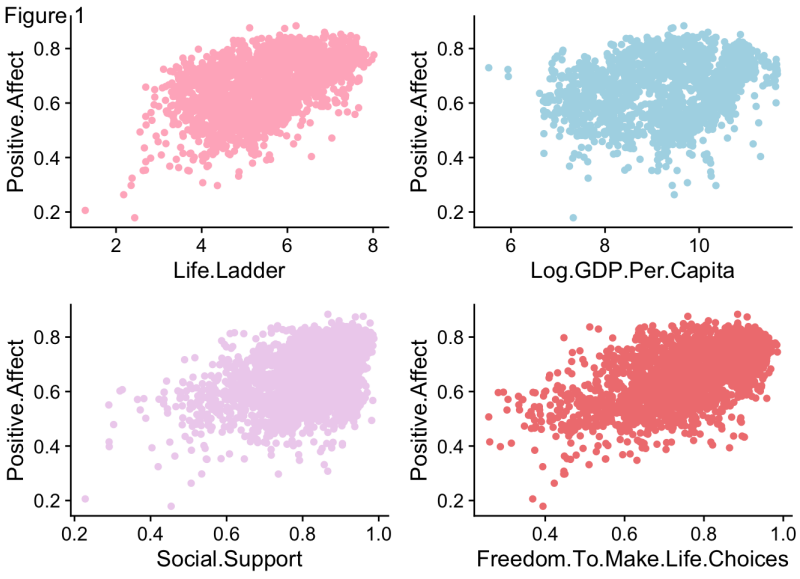
```
## Warning: Removed 24 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 35 rows containing missing values (`geom_point()`).
```

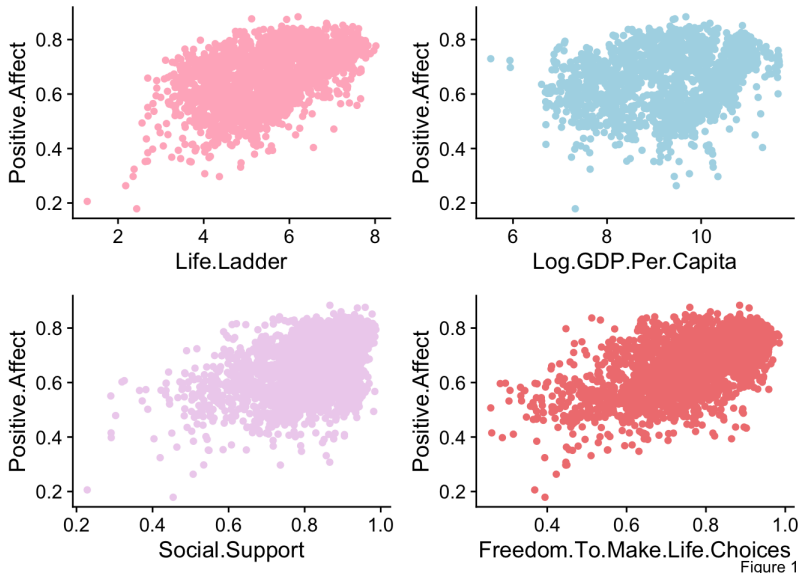
```
## Warning: Removed 26 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 48 rows containing missing values (`geom_point()`).
```

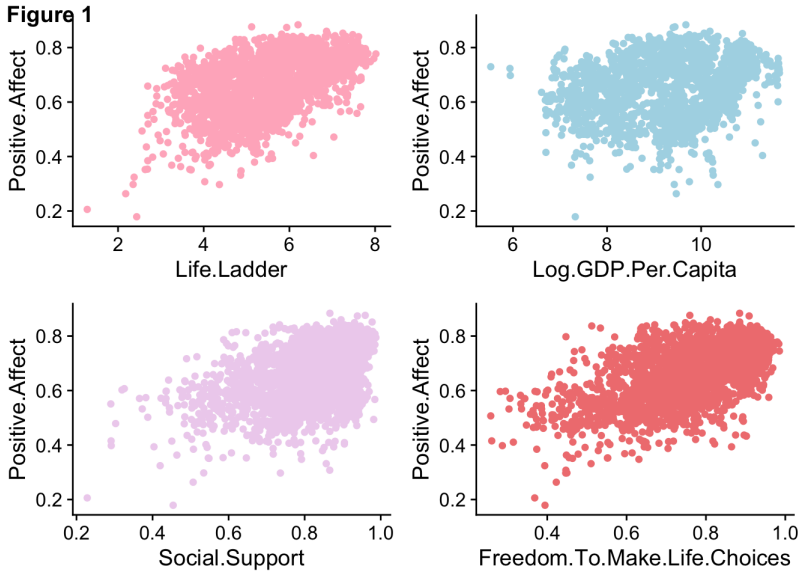
```
# Default font size and position
p + draw_figure_label(label = "Figure 1")
```



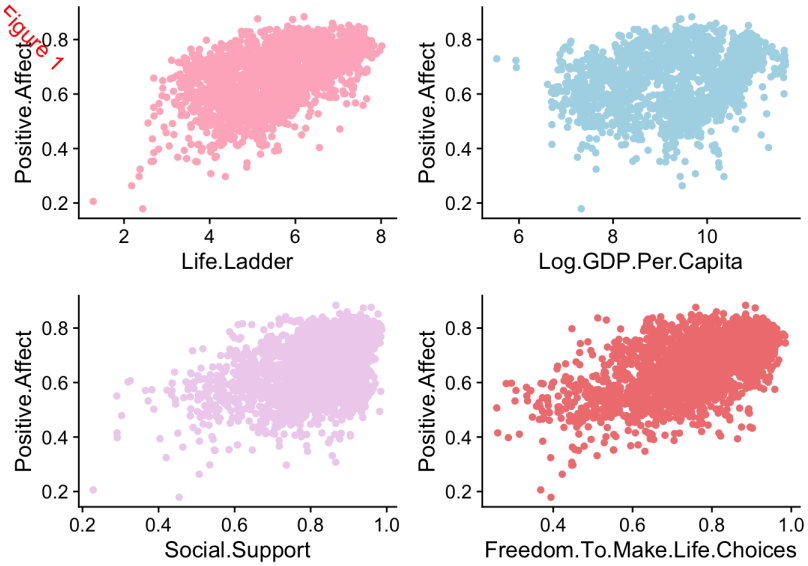
```
# Different position and font size
p + draw_figure_label(label = "Figure 1", position = "bottom.right", size = 10, )
```



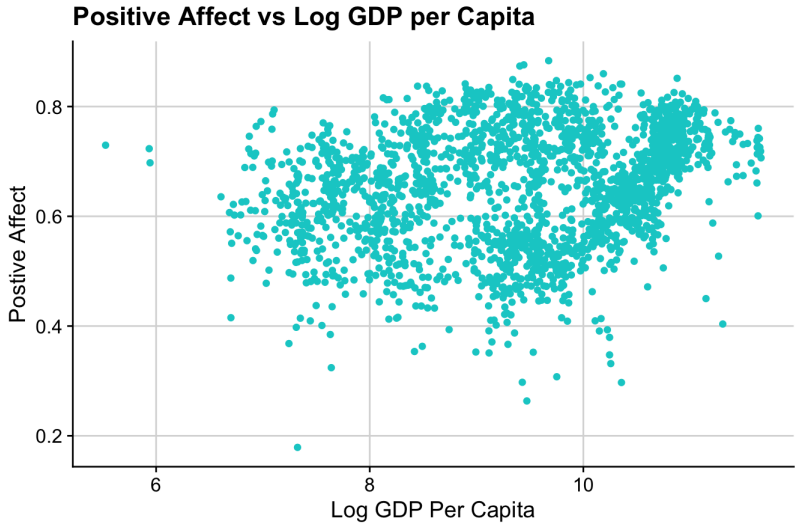

```
# Using bold font face
p + draw_figure_label(label = "Figure 1", fontface = "bold")
```




```
# Making the label red and slanted
p + draw_figure_label(label = "Figure 1", angle = -45, colour = "red")
```




```
# Labeling an individual plot
ggdraw(p2) + draw_figure_label(label = "Figure 1", position = "bottom.right", size = 10)
```

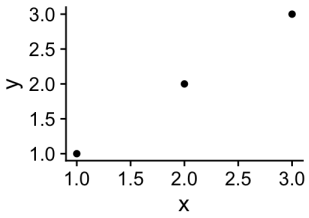


This formula has no relevance here:

Figure 1

Case 5) Drawing a subplot

```
# make a plot
p <- ggplot(data.frame(x = 1:3, y = 1:3), aes(x, y)) +
geom_point()
# draw into the top-right corner of a larger plot area
ggdraw() + draw_plot(p, .6, .6, .4, .4)
```

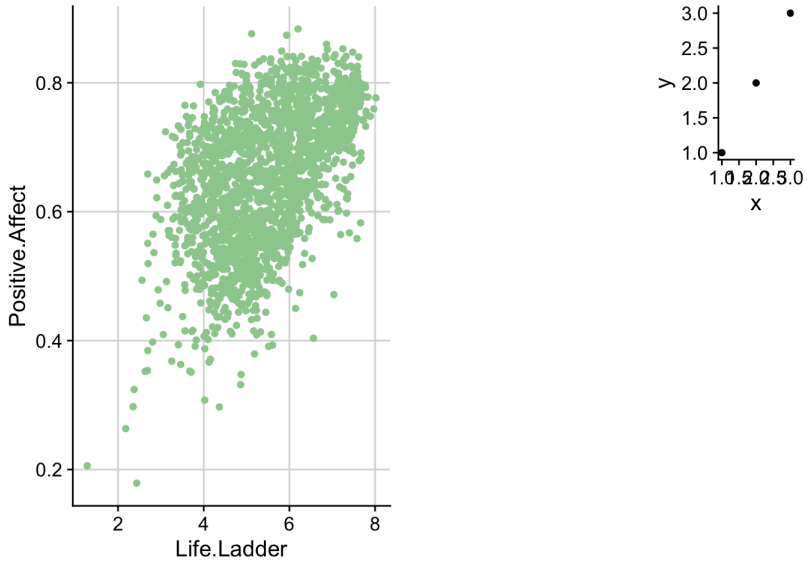


Combining with a ggplot

```
# create the first plot
plot1 <- ggplot(df, aes(Life.Ladder, Positive.Affect)) +
  geom_point(colour = "darkseagreen3") +
  theme_half_open() +
  background_grid()

# create the second plot
plot2 <- ggplot(data.frame(x = 1:3, y = 1:3), aes(x, y)) +
  geom_point()

# draw the second plot as an inset
inset_plot <- ggdraw() + draw_plot(plot2, .6, .6, .4, .4)

# combine the plots using plot_grid
combined_plot <- plot_grid(plot1, inset_plot, ncol = 2)
```

```
## Warning: Removed 24 rows containing missing values (`geom_point()`).
```

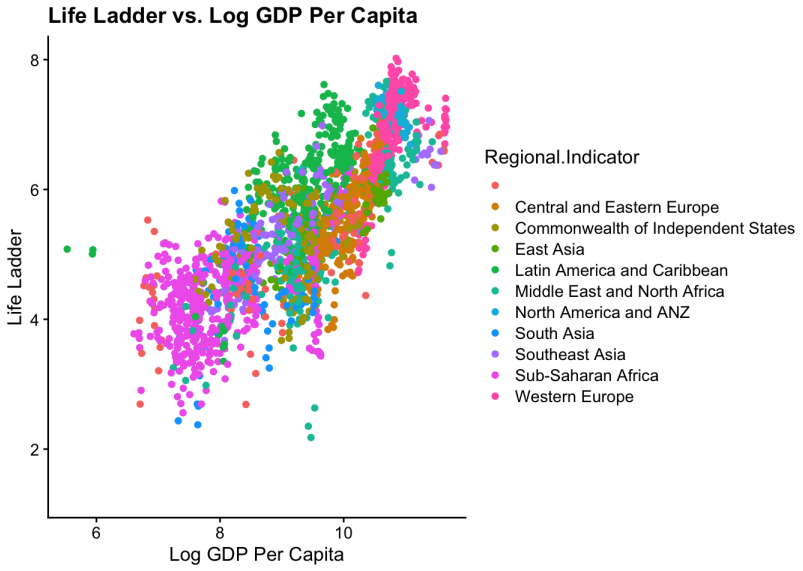
```
# display the combined plot
combined_plot
```



Case 6) Combining images and plots

```
plot <- ggplot(df, aes(x = `Log.GDP.Per.Capita`, y = `Life.Ladder`, color = `Regional.Indicator`)) +
  geom_point() +
  labs(x = "Log GDP Per Capita", y = "Life Ladder") +
  labs(title = "Life Ladder vs. Log GDP Per Capita") +
  theme_half_open(12)

plot
```

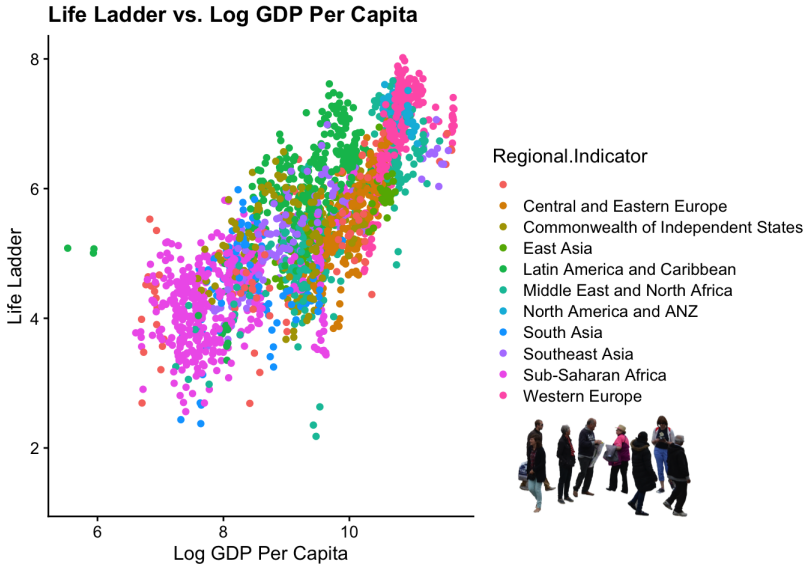
```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```



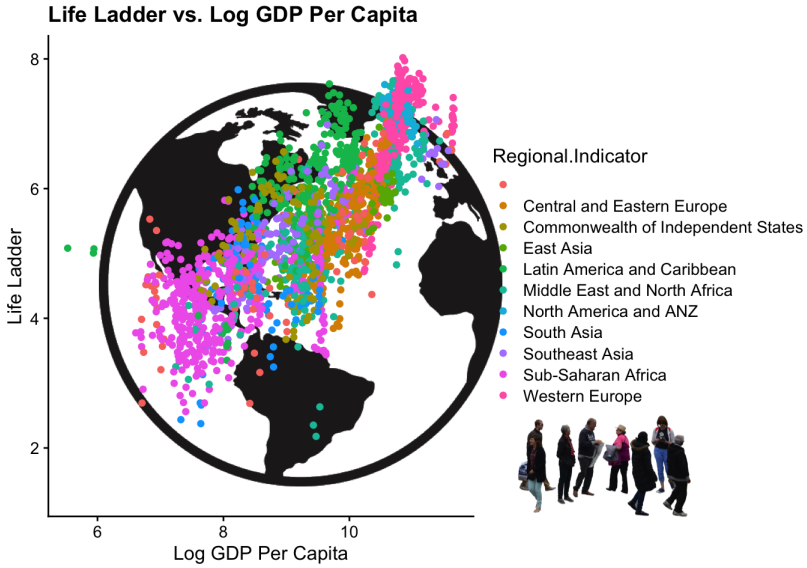

```
# Plot 2: Adding an image
plot_img <- ggdraw() +
  draw_image("people.jpg",
             scale = .3,
             x = 1,
             hjust = 1,
             halign = 0.82,
             valign = 0.05)+
  draw_plot(plot)
```

```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```

```
plot_img
```

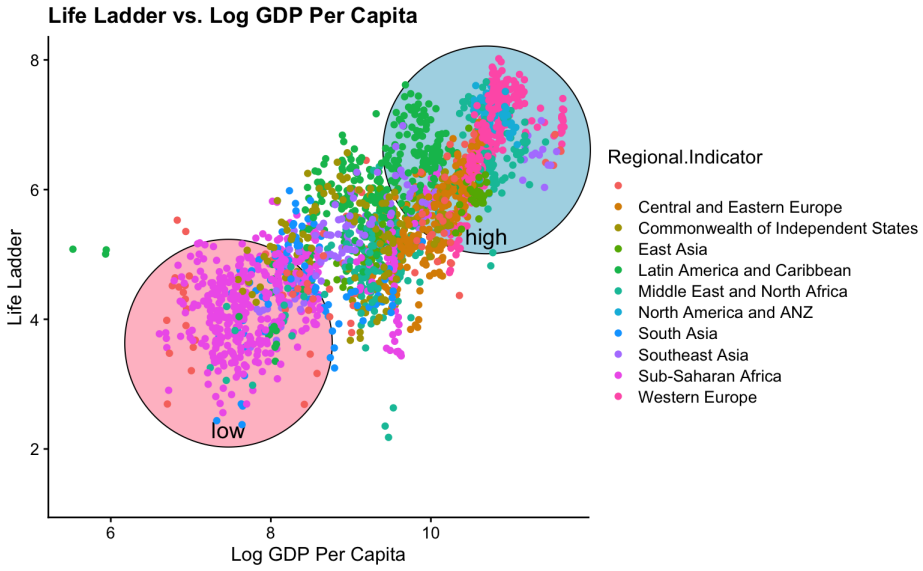



```
# Adding image as background
plot_background <- ggdraw() +
  draw_image("world.jpg",
             scale = .7,
             x = 0,
             y = 0,
             halign = 0.25) +
  draw_plot(plot_img)

plot_background
```

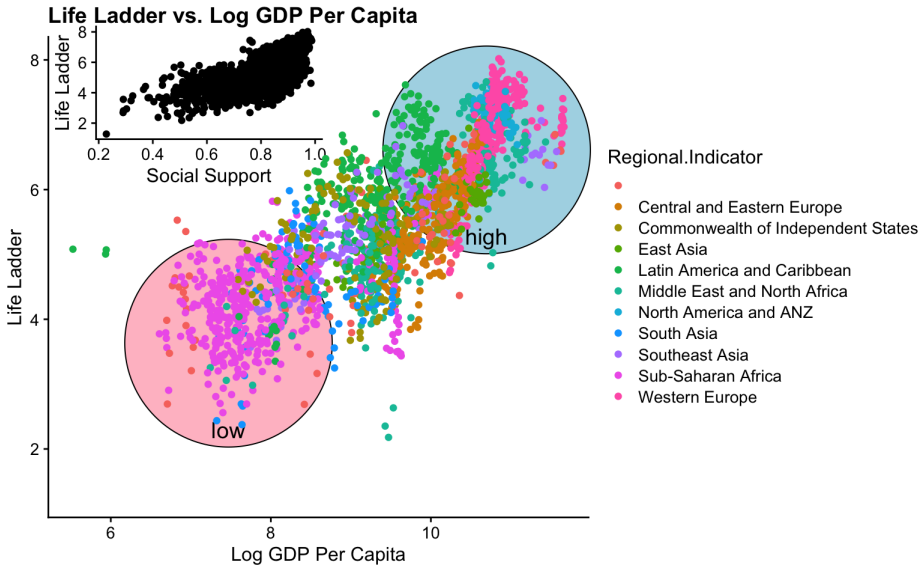


Case 7) Drawing images on plots

```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```




```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```

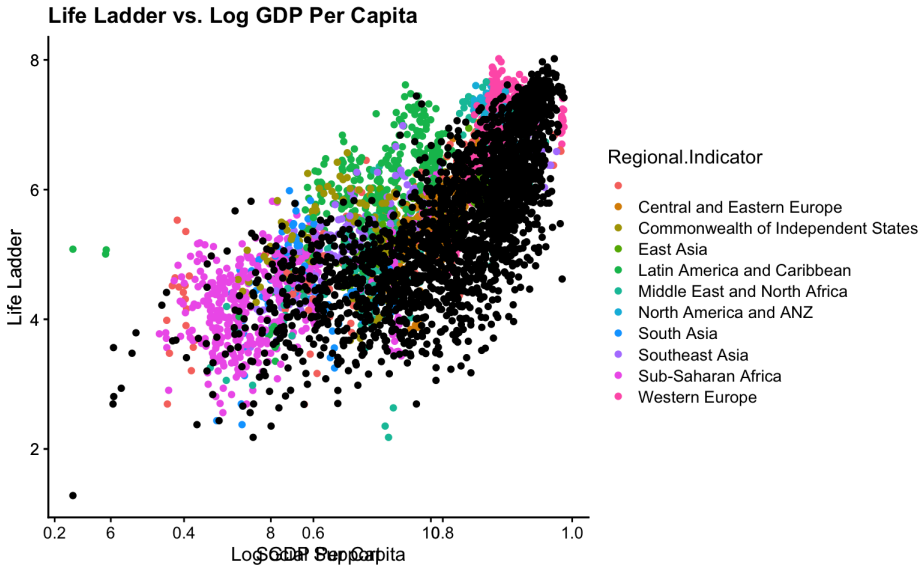


```
## Saving 8 x 5 in image
```

Case 8) Aligning plots

```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```




```
## Saving 8 x 5 in image
```