Post02

Boan Fan

12/2/2017

Motivation and Background

Have you ever felt bored with the plots produced by the “plot” function in R? Graphing is essentially the core to data analysis, and as we begin to process more complex data, the plot function is no longer powerful enough. For example, managing several graphs on the several plot while assigning different graphical features (color, width…) is not an easy task when we use the plot function.

Therefore, getting to know ggplot is very helpful. It is a powerful and flexible graphing package created by Hadley Wickham, and users can create neat, polished and organized plots in an easy way. The package is based on Leland Wilkinson’s The Grammar of Graphics and it allows the creation of graphs of univariate, multivariate and categorical data. The basic idea of The Grammar of Graphics is to independently specify plot building blocks and combine to create any kind of graphs we want.

Background

Before applying ggplot to a dataset, we need to know some functions and variables that are required to assemble a plot by ggplot. Firstly, ggplot is the main function where we specify the dataset and variables to plot. Geoms are the geometric objects to display (geom_points, geom_bar, geom_density…). And another argument, aes (short for aesthetics) specify the shape, transparency, color and line type. Note that variables are mapped to aesthetics with the aes() function, while fixed aesthetics are set outside the aes() call.
#Example

First of all, in order to use ggplot, we have to call the ggplot2 package.

```
library(ggplot2)
```

for demonstrational purpose, we will use the convenient dataset iris. It gives the measurements in centimeters of the variables sepal length and width and petal length and width for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica. The data for the first 5 flowers gives a general idea what the data set looks like

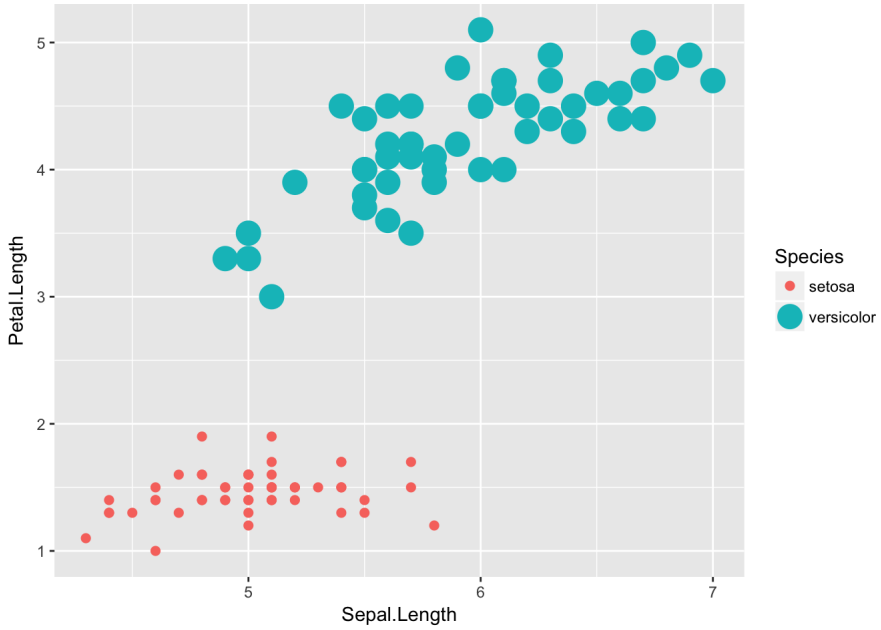
```
data(iris)
head(iris, 5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
```

One significant advantage ggplot has over regular plot is that it is much easier to handle complex graphs with ggplot. For example, we want to display the relationship sepal length and petal length for the sepcies of setosa and versicolor.

```
ggplot(subset(iris, Species %in% c("versicolor", "setosa")),
       aes(x = Sepal.Length, y = Petal.Length, color = Species, size = Species)) +
  geom_point()
```

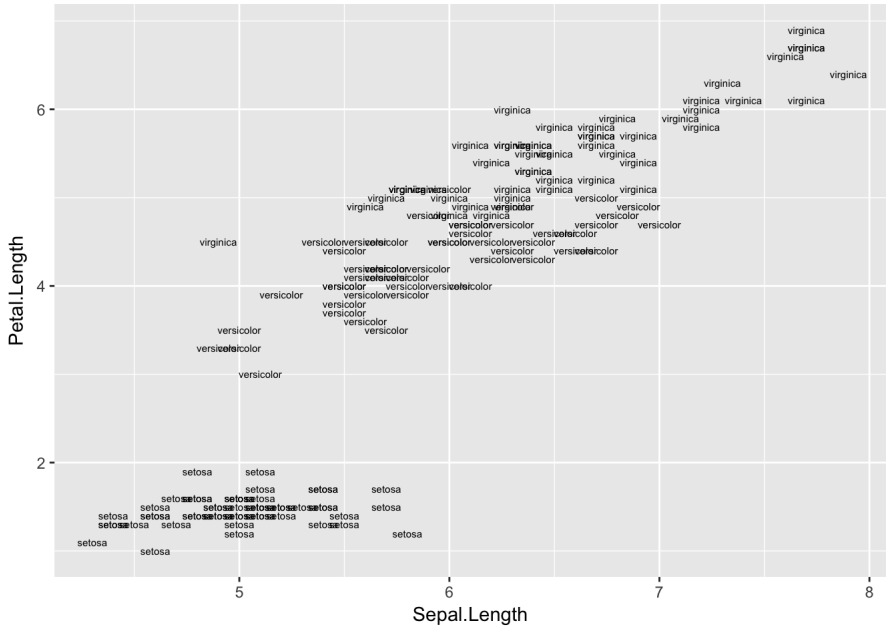
```
## Warning: Using size for a discrete variable is not advised.
```



Therefore, it is very easy to subset two groups of data and display their graphs with ggplot. Also, it is very easy to manipulate the size and colors of the subsets and make the comparison visually apparent.

Also, we can manipulate the plot in the way that makes the display best serves our purpose. For example, we want to replace the points on the graph with the text of their species name if the readers are too lazy to read the legend. In this case, we just need to call geom_text.

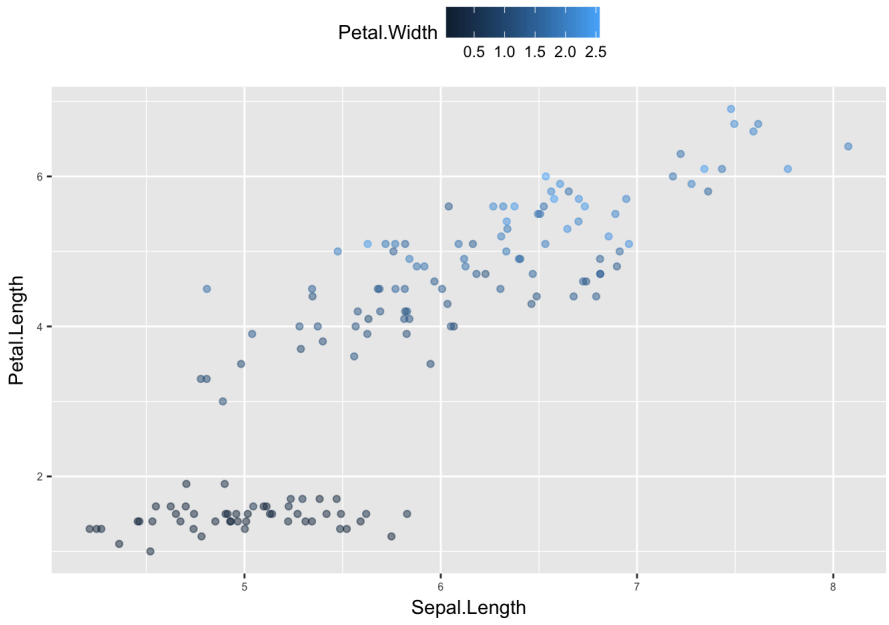
```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_text(aes(label=Species), size = 2)
```



Aesthetic mapping, in some way, is very limited. The aesthetic is directly linked to the variable, and we cannot decide what color or shapes should be used. Describing what colors/shapes/sizes is done by changing the scales, which include position, color and fill, size, shape and line type.

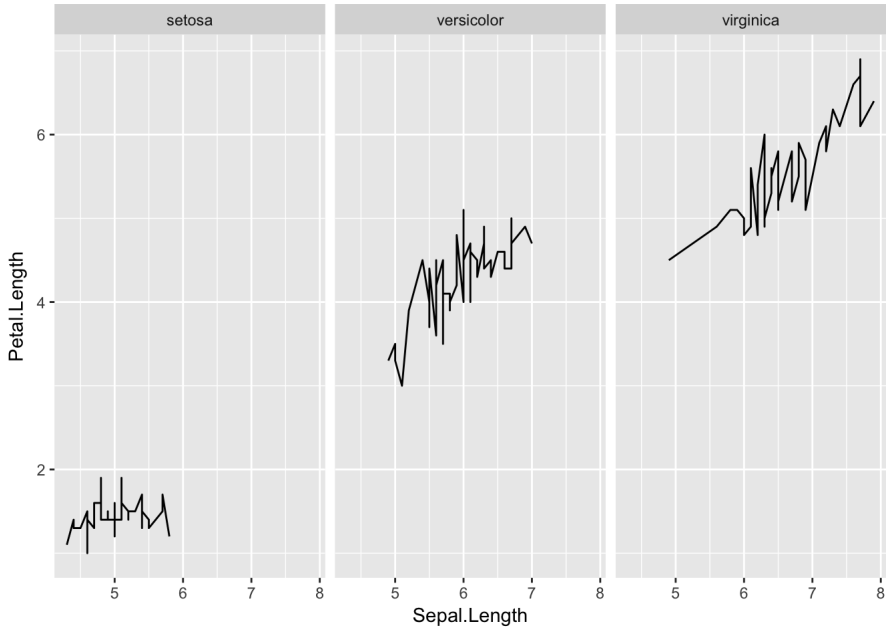
Now, we want to display a dotplot showing the distribution of sepal length by petal lenght and petal width. We also want to adjust the scales to personalize the plot.

```
p1 <- ggplot(iris,
             aes(x = Sepal.Length,
                 y = Petal.Length)) +
       theme(legend.position="top",
             axis.text=element_text(size = 6))
p2 <- p1 + geom_point(aes(color = Petal.Width),
                      alpha = 0.5,
                      size = 1.5,
                      position = position_jitter(width = 0.25, height = 0))
p2
```



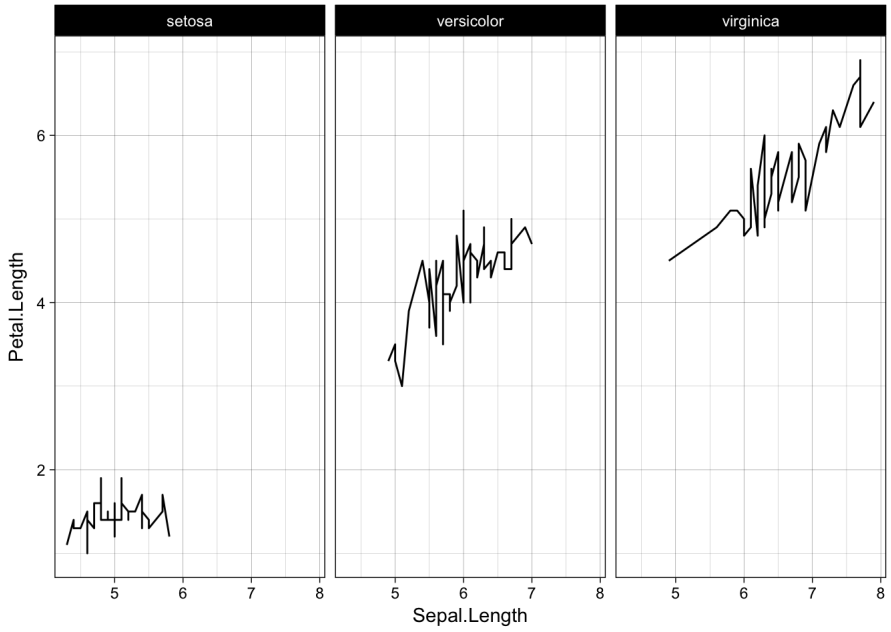
Another very useful aspect of ggplot is facet. When display the relationship sepal length and petal length for the sepcies of setosa and versicolor, the data are on the same plot. What if we want the data on different plots and displayed side by side.

```
p3 <- ggplot(iris, aes(x = Sepal.Length, y = Petal.Length))
p4 <- p3 + geom_line() +
   facet_wrap(~Species)
p4
```



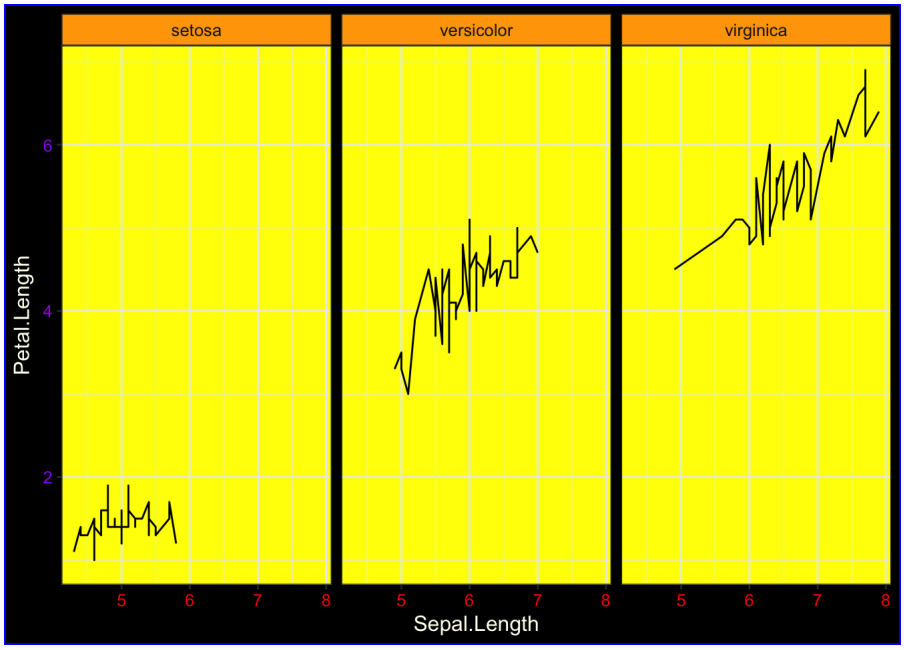
Another feature of ggplot is theme. It can dictate plot features that are irrelevant to data such as axis labels, plot background, facet label background, and legend appearance. Some of the built-in themes include theme_gray(), which is the default theme, theme_bw(), and theme_classc().

```
p4 + theme_linedraw()
```



We can also create our own theme with ggplot by overriding the default theme setting colors, texts and borderlines of our choice.

```
theme_new <- theme_bw() +
  theme(plot.background = element_rect(size = 1, color = "blue", fill = "black"),
        text=element_text(size = 12, color = "ivory"),
        axis.text.y = element_text(colour = "purple"),
        axis.text.x = element_text(colour = "red"),
        panel.background = element_rect(fill = "yellow"),
        strip.background = element_rect(fill = "orange"))
p4 + theme_new
```



Conclusion

For now, there are still some limitations of its capabilities. Some of these limitations include the inability to create a 3-dimensional graphics (which can be solved by the rgl package). Also, there are no interactive graphics that can be created with ggplot2. This problem is complemented by the ggvis package.

Despite all the limitations of this package, it is still a very powerful plotting tool in R. It can produce plots with specific details at a high level of abstraction. It provides a system to personalize and polish a plot. Also, it is very mature, used by many users.

Take Home Message

i feel the most important point to take is that ggplot2 allows us to plot in a logical way, compared to conventional plotting method. It allows much more freedom in terms of graphical features.

References

- http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html#Dumbbell%20Plot
- http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html
- http://www.sthda.com/english/wiki/ggplot2-scatter-plots-quick-start-guide-r-software-and-data-visualization
- https://www.statmethods.net/advgraphs/ggplot2.html
- https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf
- https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf
- http://ggplot2.org