Deeper Exploration in Data Visualization with ggplot2 Extensions

Yutong Song

12/2/2017

Introduction

During my exploration in data visualization with R, I find the `ggplot2` is the most useful package in this fields with many extensions to make the visualized graph more self-explanatory and characteristic. Since we only have covered the basic of `ggplot2` with some simple features, I want to use this post to explore with you some extensions of `ggplot2` to enrich and enhance our knowledge in data visualization. In this post, I am going to explore in four `ggplot2` extensions: 1. `ggrepel` 2. `ggthemes` 3. `ggforce`

Package Installation and Preparation

Pacakges `ggtheme`, `ggforce`, `ggrepel` along with `ggplot2` can be installed directly installed from CRAN:

```
install.packages("ggplot2")
install.packages("ggthemes")
install.packages("ggforce")
install.packages("ggrepel")
```

The last step is to load all the packages that we have installed:

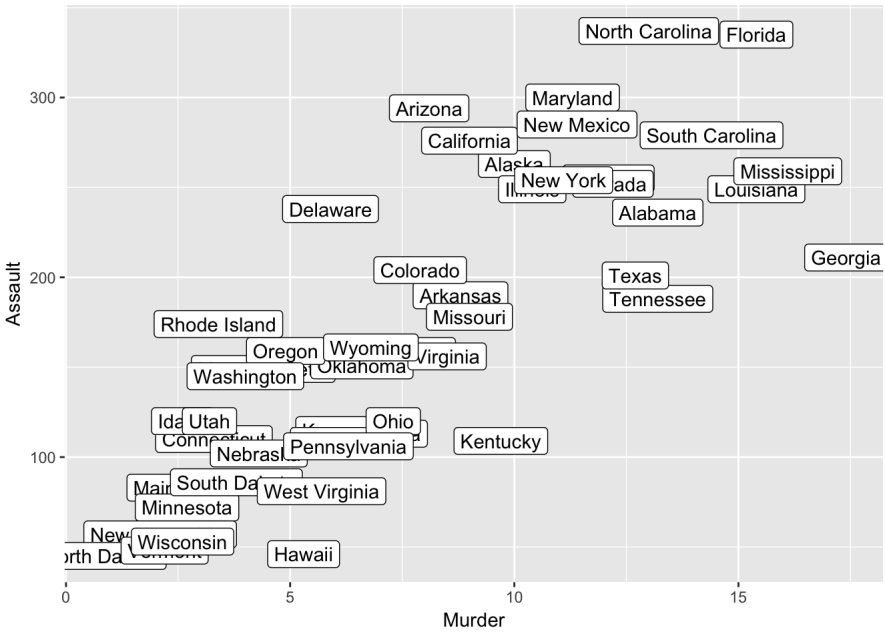
```
library(ggplot2)
library(ggforce)
library(ggthemes)
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 3.4.2
```

ggrepel: Overlapping Labels Separation and Manipulation

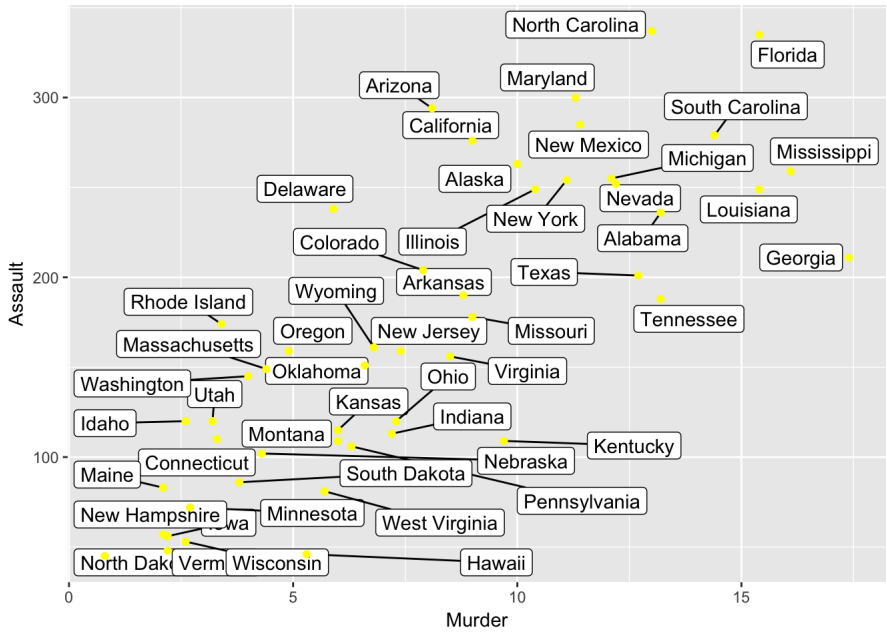
The package `ggrepel` comes in handy when we are dealing with many overlapping labels. To illustrate my points, let use the built-in data set `USArrests` (the criminal record in the US) as an example. Suppose we want to create a scatter plot of Murder and Assault with the States as the labels:

```
# create a scatter plot of Murder and Assault with the States as the labels
ggplot(dat = USArrests, aes(x = Murder, y = Assault)) +
  geom_label(aes(label = rownames(USArrests)))
```



We can see from the scatter plot above that some of the labels are overlapping, and we can resort to `ggrepel` to repel overlapping labels:

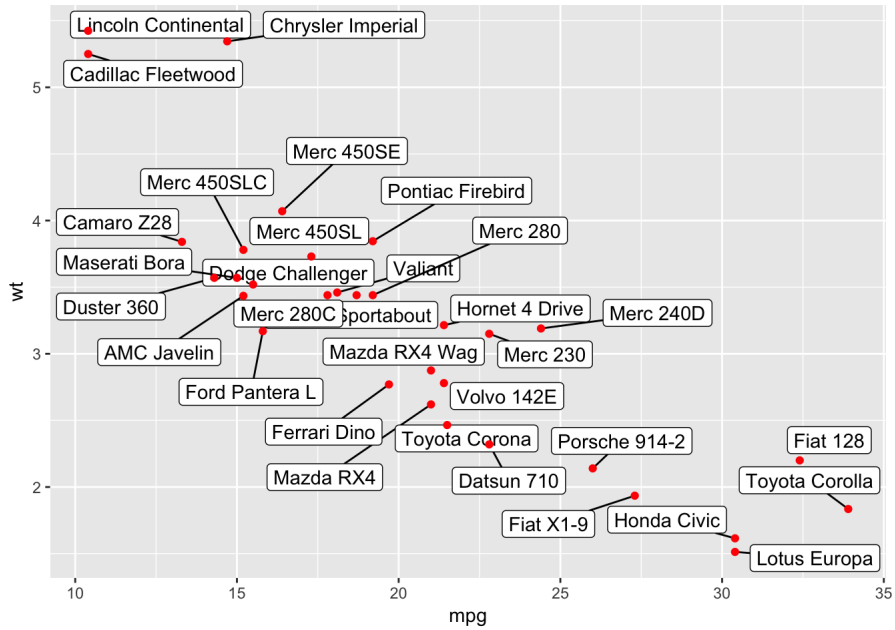
```
# create a scatter plot of Murder and Assault with the States as the labels with no overlapping labels
# using the argument geom_label_repel instead of geom_label achieves the label separation
ggplot(dat = USArrests, aes(x = Murder, y = Assault)) +
  geom_label_repel(aes(label = rownames(USArrests))) + geom_point(color = 'yellow')
```



Now we can see that the labels are separated with some lines connecting point to label.

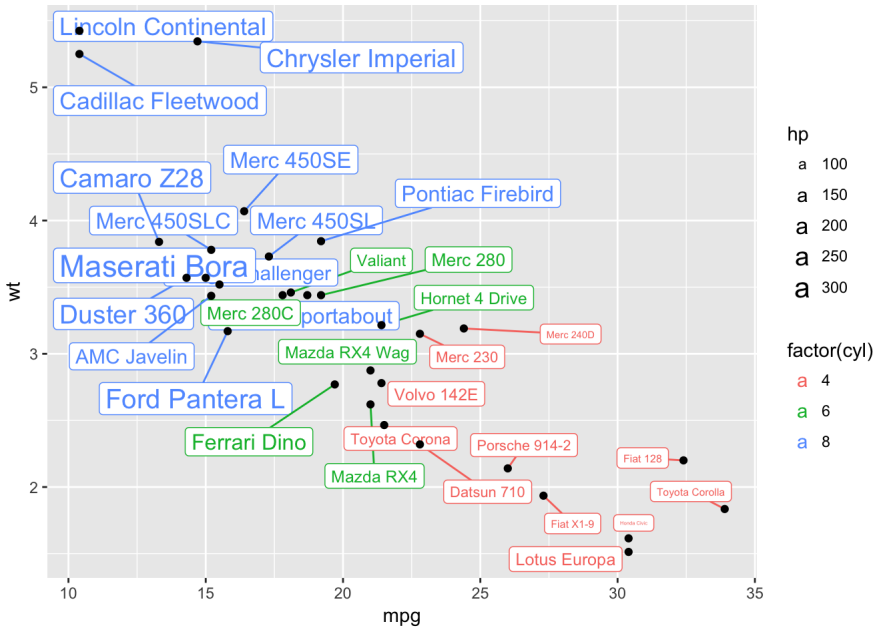
Moreover, we can manipulate the separated labels to enrich a scatter plot with more information. We can use the built-in data set `mtcars` (Motor Trend Car Road Tests) as an example. Suppose we want to create a scatter plot of mpg: Miles/(US) gallon and wt: Weight (1000 lbs) with separated labels:

```
# create a scatterplot of mpg and wt
ggplot(dat = mtcars, aes(x = mpg, y = wt)) +
  geom_label_repel(aes(label = rownames(mtcars))) + geom_point(color = 'red')
```



We can manipulate the label and points so that the size of the label are proportional with hp: Gross horsepower, and the color of labels are associated with their cyl: Number of cylinders:

```
# change size and color of labels by changing the color and size argument in geom_label_repel
ggplot(dat = mtcars, aes(x = mpg, y = wt)) +
  geom_label_repel(aes(label = rownames(mtcars), color = factor(cyl), size = hp)) + geom_point(color = 'black')
```



ggthemes: Aesthetic Aspect of Data Visualization

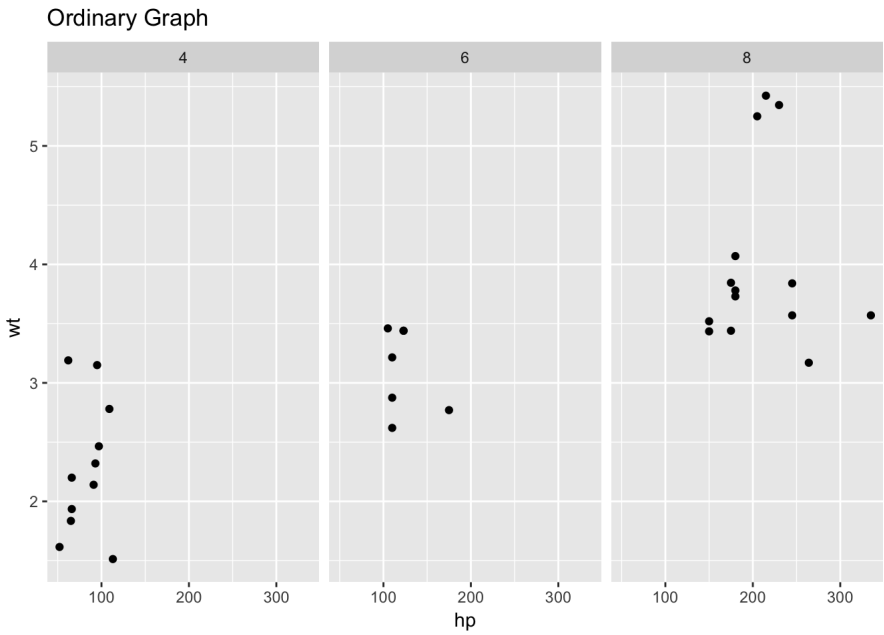
Despite the correctness of a graph, the package `ggthemes` focuses on adding aesthetic identity to your graph. With `ggthemes`, we can change the color and theme of our graph.

To change the theme of a graph, there are many functions in `ggthemes`, such as:

- `theme_economist()` : change plots to the theme as identical as them in The Economist magazine
- `theme_gdocs()` : change plots to the theme as identical as them in Google Docs
- `theme_wsj()` : change plots to the theme as identical as them in Wall Street Journal

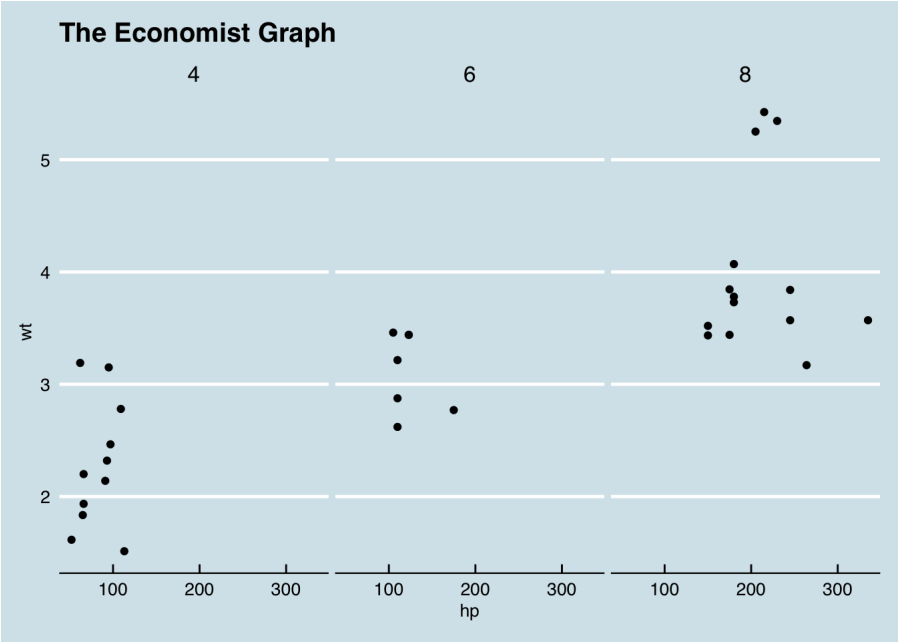
Not only that, some of them have matching functions `scale_colour_economist()`, `scale_colour_gdocs()`, `scale_colour_wsj()` and to change the color of the graph as well. Take the `mtcars` as an example. Suppose we want to create scatter plot of hp: Gross horsepower and wt: Weight (1000 lbs) grouped by cyl: Number of cylinders:

```
# scatterplot by cyl
graph <- ggplot(data = mtcars, aes(x = hp, y = wt)) +
  geom_point() +
  facet_wrap(~ cyl)
graph +
  ggtitle("Ordinary Graph")
```

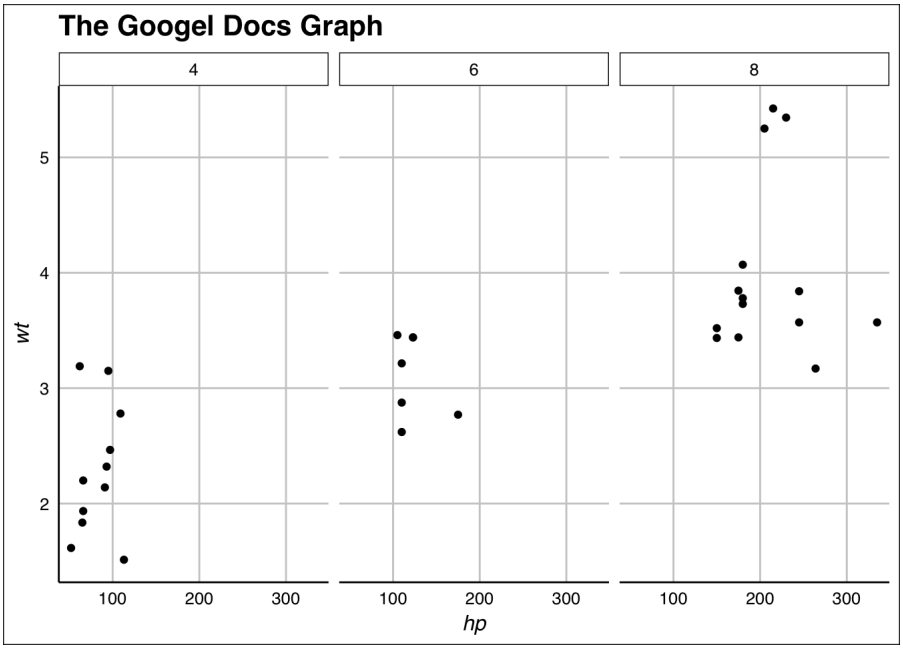


We can change the style of this graph dramatically using the functions I list above:

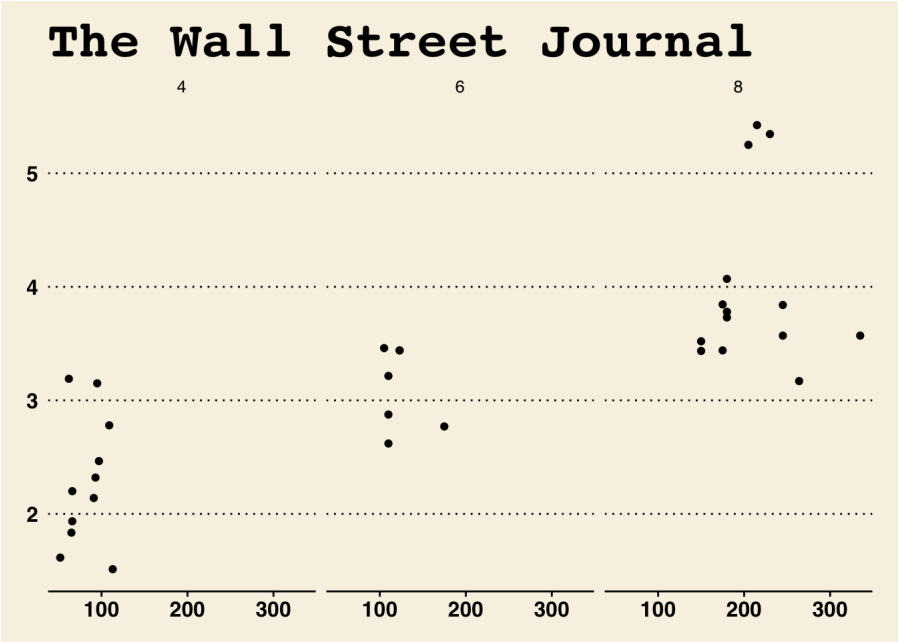
```
# scatterplot by cyl of The Economist theme
# use theme_economist and scale_colour_economist to specify the style
graph +
  theme_economist() +
  scale_colour_economist() +
  ggtitle("The Economist Graph")
```




```
# scatterplot by cyl of The Google Docs theme
# use theme_gdocs and scale_colour_gdocs to specify the style
graph +
  theme_gdocs() +
  scale_colour_gdocs() +
  ggtitle("The Googel Docs Graph")
```



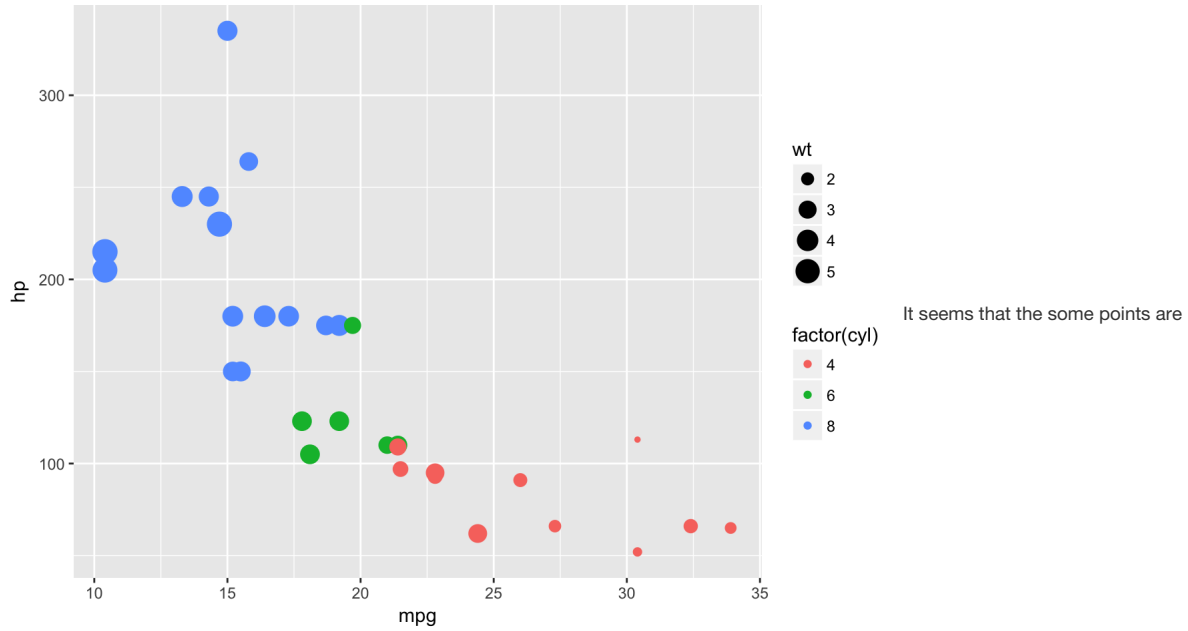

```
# scatterplot by cyl of The Wall Street Journal theme
# use theme_wsj and scale_colour_wsj to specify the style
graph +
  theme_wsj() +
  scale_colour_wsj() +
  ggtitle("The Wall Street Journal")
```



ggforce: Capacity Enhancement of ggplot2

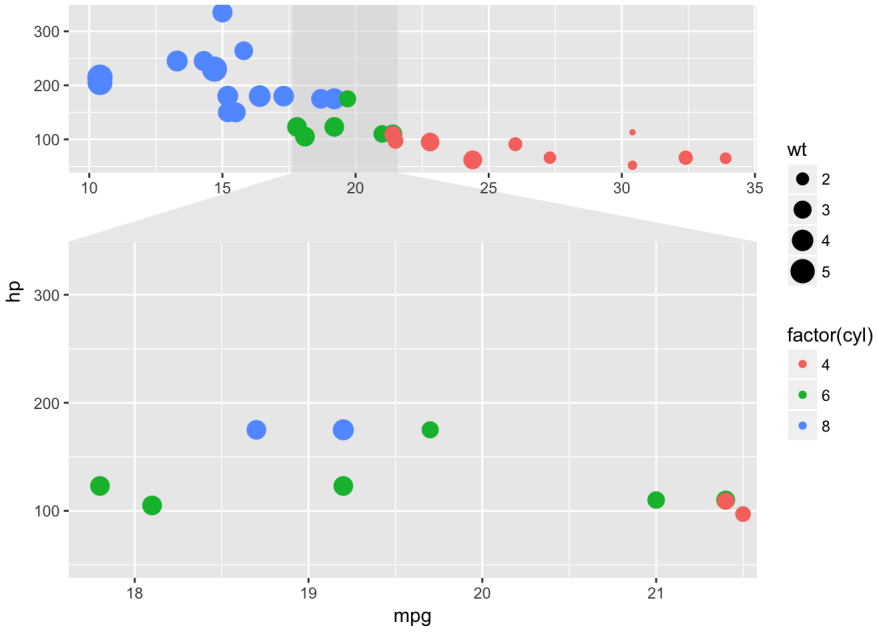
The Package `ggforce`, as its names, brings a overall development to the capacity of `ggplot2` with many extension arguments. In this lab, I want to introduce one functionality that `ggforce` enables `ggplot2`: Zooming. Let us use `mtcars` as an example. Suppose we want to create create a scatterplot of mpg and hp:

```
# scatterplot of mpg and hp with color showing cyl and size showing wt
ggplot(dat = mtcars, aes(x = mpg, y = hp, color = factor(cyl), size = wt)) +
  geom_point()
```

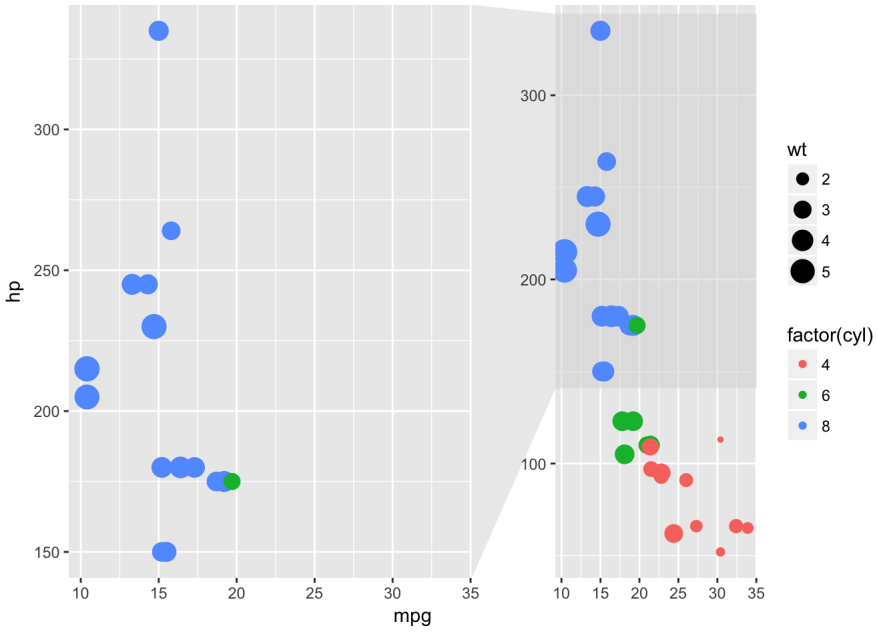


It seems that the some points are more concentrated, and we want to study the spread of cars more closely, we can use the `facet_zoom()` provided by `ggforce` to zoom in either horizontally, vertically, or together.

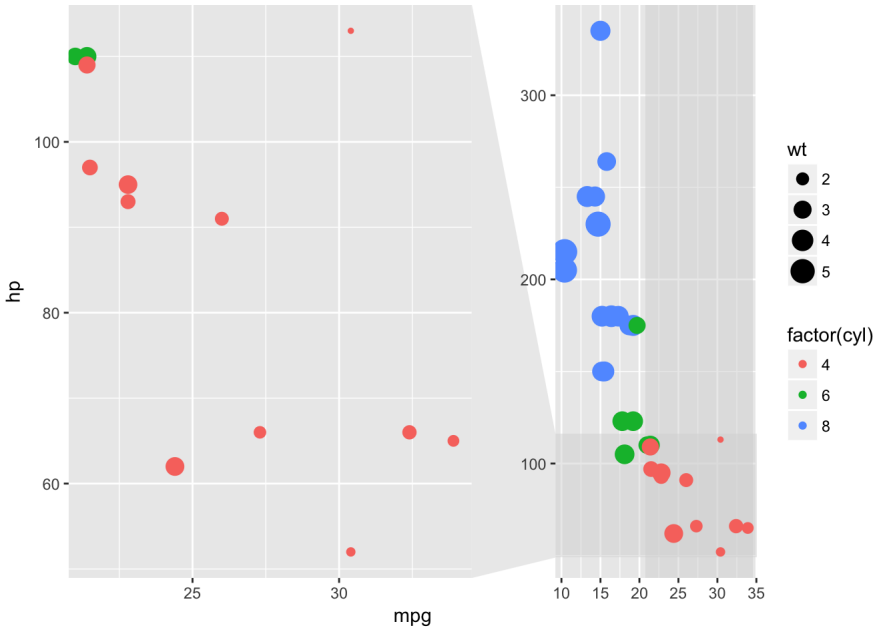
```
# zoom in on cars with cyl of 6 horizontally
ggplot(dat = mtcars, aes(x = mpg, y = hp, color = factor(cyl), size = wt)) +
  geom_point() +
  facet_zoom(x = cyl == 6) # notice that by setting x = cyl == 6 we specify that we want to zoom in the x axis
```




```
# zoom in on cars with cyl of 8 vertically
ggplot(dat = mtcars, aes(x = mpg, y = hp, color = factor(cyl), size = wt)) +
  geom_point() +
  facet_zoom(y = cyl == 8) # notice that by setting y = cyl == 8 we specify that we want to zoom in the y axis
```




```
# zoom in on cars with cyl of 4 horizontally and vertically
ggplot(dat = mtcars, aes(x = mpg, y = hp, color = factor(cyl), size = wt)) +
  geom_point() +
  facet_zoom(xy = cyl == 4)
```




```
# notice that the zoom is both horizontal and vertical simultaneously
```

Take Home Message

In this post, we have explored three powerful ggplot extension packages: `ggrepel` can separate overlapping labels in a graph; `ggthemes` enables us to change the theme and color of a graph; and `ggforce` develops overall capacity of `ggplot2` such as zoom in a graph from a specific axis.I hope my post can motivate you into exploring more in to data visualization with `ggplot2`.

References

1. https://cran.r-project.org/web/packages/ggforce/vignettes/Visual_Guide.html#contextual-zoom
2. http://www.ggplot2-exts.org/ggforce.html
3. http://www.ggplot2-exts.org/ggrepel.html
4. http://www.ggplot2-exts.org/gallery/
5. https://github.com/jrnold/ggthemes
6. https://github.com/slowkow/ggrepel
7. https://cran.r-project.org/web/packages/ggrepel/vignettes/ggrepel.html
8. https://cran.r-project.org/web/packages/ggforce/ggforce.pdf