Advanced graphics: practical 2 solutions

Dr Colin S. Gillespie

This practical aims to guide you through some of the key ideas in `ggplot2`. As with the first practical, feel free to experiment. Some of the functions introduced in this practical haven't been explicitly covered in the notes. Use the built-in R help or the `ggplot2` help pages at `http://had.co.nz/ggplot2/` as needed.

1 *Introduction*

To begin, load the library:

```
library("ggplot2")
```

and the `mpg` data set

```
data(mpg)
dim(mpg)
```

2 *Basic plots*

The aim of this section is to recreate the graphics in figure 1. Feel free to experiment.

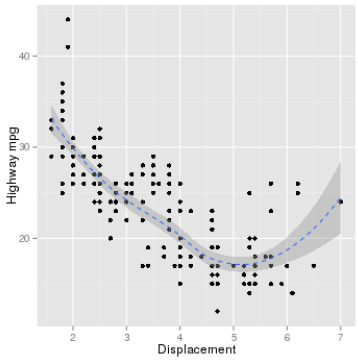

Figure 1: Graphics for section 1.

1. Figure 1: Create a scatter plot of engine displacement, `displ`, against highway mpg, `hwy`. To get started:

   ```
   ggplot(data=mpg, aes(x=displ, y=hwy)) +
     geom_point() + xlab("Displacement")
   ```

   Now add a dashed loess line and change the y-axis label. Hint: try `stat_smooth` and `xlab('New label')`.

   ```
   g = ggplot(data=mpg, aes(x=displ, y=hwy))
   ```

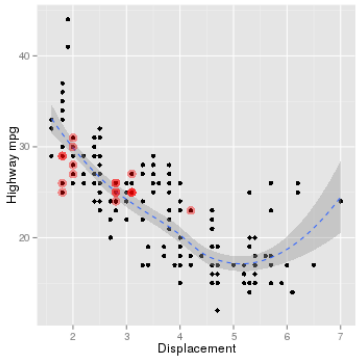

Figure 2: Graphics for section 1.

2. Figure 2: highlight the Audi cars with a slightly larger red circle.

   ```
   g1 = g + geom_point() + stat_smooth(linetype=2) +
     xlab("Displacement") + ylab("Highway mpg")
   ```

3. Figure 3: Using `stat_smooth`, add a loess line conditional on the drive.

   ```
   g2 = g1 + geom_point(data=subset(mpg, manufacturer=="audi"),
      aes(x=displ, y=hwy),
     colour="red", size=4, alpha=0.4)
   ```

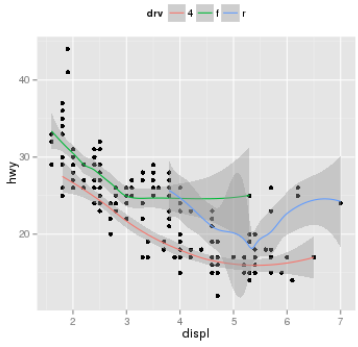

Figure 3: Graphics for section 1.

3 *Over plotting*

Scatter plots are very useful. However, when we have a large data set, points will be plotted on top of each other obscuring the relationship. We call this problem over plotting. There are a few techniques we can use to help, although the best solution is often problem specific.

To begin with we will create an example data frame:

```
## If your computer is slow when plotting
## reduce the value of n
n = 20000
x1 = signif(matrix(rnorm(n), ncol=2), 2)
x2 = signif(matrix(
  rnorm(n, mean=3, sd=1.5), ncol=2), 2)
x = rbind(x1, x2)
df = data.frame(x=x[,1], y=x[,2])
```

We can create a simple scatter plot of this data using the following commands:

```
h= ggplot(df) + geom_point(aes(x, y))
```

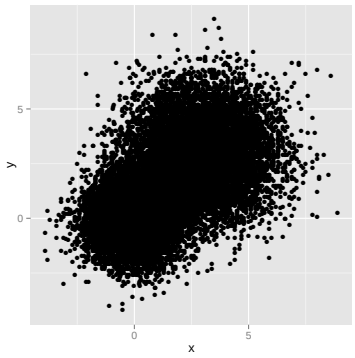

Figure 4: A scatter plot that suffers from over plotting.

This plot isn't particularly good. Try to improve it by using a combination of:

- changing the transparency level: `alpha`;
- change the shape: `shape=1` and `shape='.'`
- use some jittering - `geom_jitter`.
- adding a contour to the plot using `stat_density2d`.
- What does

  ```
  h +  stat_density2d(aes(x,y, fill=..density..),
                  contour=FALSE, geom="tile")
  ```

  do?
- What does `stat_bin2d()` and `stat_binhex()` do[1] - add it to the plot to find out! Try varying the parameters `bins` and `binwidth`.

[1] To use `stat_binhex` you may need to install the `hexbin` package.

4 *Displaying distributions*

The diamonds data set contains the prices and other attributes of almost 54,000 diamonds. It is a data frame with 53,940 rows and 10 variables. First, load the diamonds data set:

```
data(diamonds)
```

and look at the help file:

```
?diamonds
```

We can construct a histogram of diamond depth using the following commands:

```
i1 = ggplot(data=diamonds) +
  geom_histogram(aes(x=depth))
```

to get figure 5. Let's experiment a bit.

1. Change the `binwidth` in the `geom_histogram`. What value do you think is best?
2. What happens when you set `colour=cut` in the `geom_histogram` aesthetic? What other options can you change?[2]
3. Try `geom_density`. Set `fill=cut` and change the `alpha` value.
4. Try `geom_boxplot`.

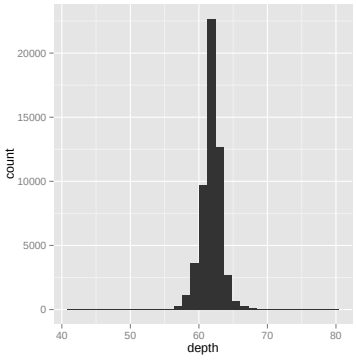

Figure 5: Histogram of the diamond data set.

[2] Look at the `geom_histogram` help page: `http://had.co.nz/ggplot2/geom_histogram.html`

Solutions

Solutions are contained within this package:

```
library(nclRggplot2)
vignette("solutions2", package="nclRggplot2")
```