Practical 2

Jumping Rivers

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.

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
library("jrGgplot2")
library("ggplot2")
df = overplot_data(n = 20000)
```

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

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

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

- changing the transparency level: `alpha`;[1]
- 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 - add it to the plot to find out! Try varying the parameters `bins` and `binwidth`.

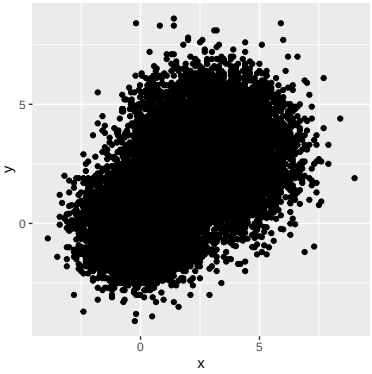

Figure 1: A scatter plot that suffers from over plotting.

[1] `alpha` takes a value between 0 and 1.

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, package = "ggplot2")
```

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 2. 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?^[Look at the `geom_histogram` help page: `http://had.co.nz/ggplot2/geom_histogram.html`}
3. Try `geom_density`. Set `fill=cut` and change the `alpha` value.
4. Try `geom_boxplot`.

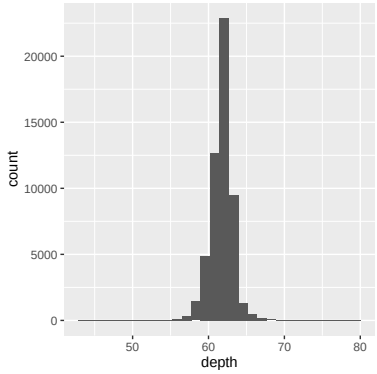

Figure 2: Histogram of the diamond data set.

Copy cat

The aim of this section is to recreate the graphics in figures 3 and 4. Feel free to experiment. To begin, load the package

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

and the `mpg` data set

```
data(mpg, package = "ggplot2")
dim(mpg)
```

1. Figure 3: 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 `ylab(New label')`‘.

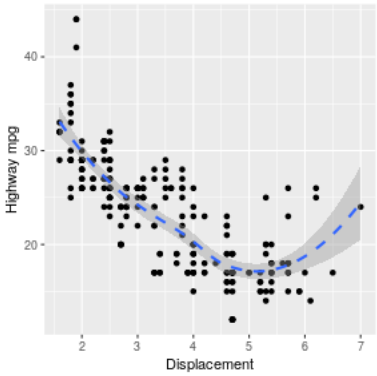

Figure 3: Graphics for section 1.

```
g = ggplot(data = mpg, aes(x = displ, y = hwy))
g1 = g + geom_point() + stat_smooth(linetype = 2) + xlab("Displa
```

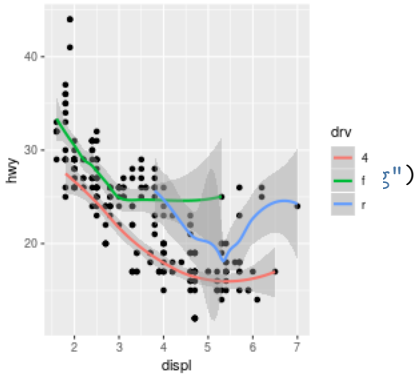

Figure 4: Graphics for section 1.

2. Figure 4: Using `stat_smooth`, add a loess line conditional on the drive.

```
g2 = g + geom_point() + stat_smooth(aes(colour = drv))
```

Solutions

Solutions are contained within this package:

```
library(jrGgplot2)
vignette("solutions2", package = "jrGgplot2")
```