

# Chapter 3 Practical

## *Jumping Rivers*

### Getting Started

First load the **ggplot2** package

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

and the OK Cupid data set

```
data(okcupid, package = "jrTidyverse")
```

### Scatter plots

1. Let's start by creating a basic scatter plot of the heights and ages. We do this using the `geom_point()` command

```
# alpha makes the points transparent
ggplot(data = okcupid) +
  geom_point(aes(x = age, y = height), alpha = 0.2)
```

2. To save typing, we're going to store the original ggplot object as a variable

```
g = ggplot(data = okcupid)
g1 = g + geom_point(aes(x = age, y = height), alpha = 0.2)
```

So now running `g1` will produce the graph

```
g1
```

3. The arguments `x` and `y` here are called aesthetics. What do you think happens if you omit the `y` aesthetic?
4. For `geom_point()`, both the `x` and `y` aesthetics are required. This particular geom has other aesthetics: shape, colour, size and alpha.<sup>1</sup> For instance we can specify that we wish to map the variable `sex` to a colour aesthetic by including it inside `aes()`

```
g + geom_point(aes(x = age, y = height, colour = sex))
```

5. Change `colour = sex` to `colour = height`. Why do you think there's a change in the legend?

### Bar plots

`geom_bar()` can be used to create a bar chart. It requires only one aesthetic and that is `x`. For the provided aesthetic, the frequencies will be calculated and shown as a bar. For example

```
g + geom_bar(aes(x = body_type))
```

1. Change the axis labels to "Body Type" and "Total" by adding two more layers, using `+ xlab("Body Type")` and `+ ylab("Total")`
2. Split the graph up into the two genders (hint: use `colour` and `fill`).

---

<sup>1</sup>These are available for most geoms. For a collection of aesthetics see the relevant help pages.

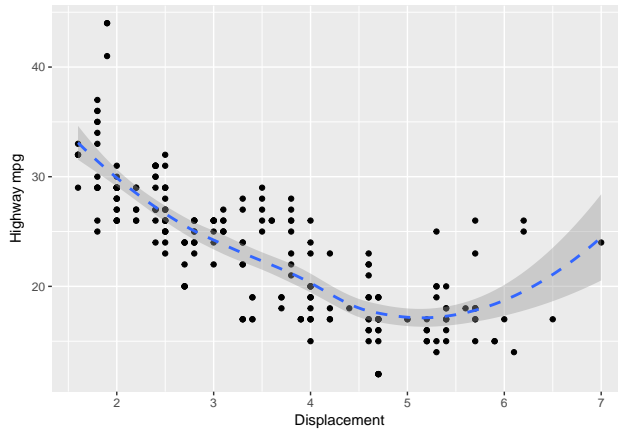


Figure 1: Graphics for section 1

3. With such long labels, it might make more sense to rotate the graph such that the bars and labels are horizontal. Switch the x and y axes using a `coord_flip()` layer added to your graph.
4. I am not too keen on how the female and male bars are displayed on top of each other. The argument to change these is `position`. The default is `stack`, for example we can put the bars next to each other using

```
g + geom_bar(aes(x = body_type, colour = sex, fill = sex), position = "dodge") +
  coord_flip()
```

Other values you might try here are `position = fill`, `position = jitter`, or `position = identity`.

5. What does the `fill` position argument do?

## Box plots

1. Box plots are a great way to visualise the shape of a distribution of some variable. Start by creating a boxplot of peoples ages in the `okcupid` data. The `x = 1` in the code below lets us have just a single boxplot for all ages

```
g = ggplot(okcupid)
g + geom_boxplot(aes(x = 1, y = age))
```

2. Switch out the `x = 1` for `x = smokes` to get a boxplot for each group

## Copy cat

The aim of this section is to recreate the graphics in figure @ref(fig:F1). Feel free to experiment. To begin, load the package

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

and the `mpg` data set

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

- 1) Figure @ref("fig:F1"): Create a scatter plot of engine displacement, `displ`, against highway mpg, `hwy`. To get started:

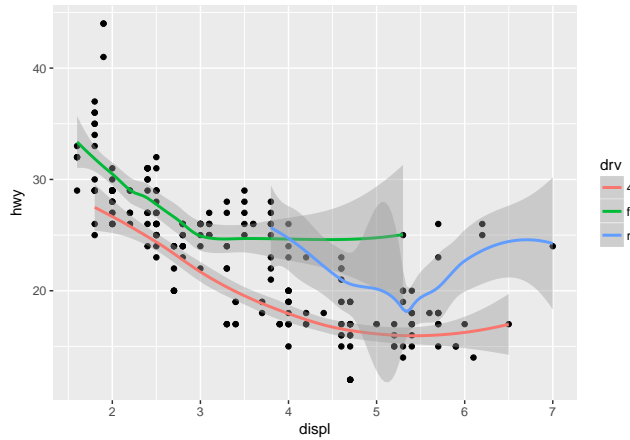


Figure 2: Graphics for section 1

```
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')`.

2) Figure @ref("fig:F2"): Using `stat_smooth()`, add a loess line conditional on the drive.