Practical 2

Jumping Rivers

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

Load the bond data set.

```
bond = readRDS(file = "data/bond.rds")
```

Scatter plots

Let's start with a basic scatter plot

```
ggplot(data = bond, aes(x = Kills, y = Alcohol_Units)) +
  geom_point()
```

The arguments `x` and `y` are called aesthetics. For `geom_point`, these parameters are required. This particular geom has other aesthetics: `shape`, `colour`, `size` and `alpha`.[1] Experiment with other aesthetics. For example,

[1] These aesthetics are usually available for most geoms.

```
ggplot(data = bond, aes(x = Kills, y = Alcohol_Units)) +
  geom_point(aes(colour = Actor))
```

or

```
ggplot(data = bond, aes(x = Kills, y = Alcohol_Units)) +
  geom_point(aes(colour = Actor,
                 shape = Nationality),
             size = 3)
```

Remember

- If you are defining an aesthetic by a column of the data it needs to go **inside** `aes()`. E.g. `geom_point(aes(colour = Actor))`
- Otherwise, define it **outside** `aes()`. E.g. `geom_point(colour = "red", size = 2)`

Some aesthetics, like `shape` must be discrete. If you try and use `shape = Relationships` you'll get an error.

- What happens if you colour the points by `Relationships`?

```
ggplot(data = bond, aes(x = Kills, y = Alcohol_Units)) +
  geom_point(aes(colour = Relationships))
```

Combining plots

The key idea with **ggplot2** is to think in terms of layers not in terms of plot "types". For example,

```
ggplot(bond, aes(x = Actor, y = Kills, colour = Nationality)) +
        geom_boxplot() +
        geom_point(size = 3)
```

- What happens to the plot if you swap the order of the `geom_boxplot()` and `geom_point()` function calls?
- Notice that we defined the `aes(colour = Nationality)` in the `ggplot()` function. This means that the colouring is inherited by **both** `geom_boxplot()` and `geom_point()`. Try removing `colour` from `ggplot()` and defining just in `geom_boxplot(colour = "Nationality")`. What happens?

Positions and flips

The bar geom has the following aesthetics: `x`, `colour`, `fill`, `size`, `linetype`, `weight` and `alpha`. Here is a command to get started:

```
ggplot(bond, aes(x = Nationality)) + geom_bar()
```

- As before, try different aesthetic combinations.

```
ggplot(bond, aes(x = Nationality)) + geom_bar(aes(fill = Actor))
```

We can adjust the layout of this bar plot using ggplot's position adjustments. The five possible adjustments are listed in table 1. The **default** adjustment is `stack`

```
ggplot(bond, aes(x = Nationality)) +
    geom_bar(aes(fill = Actor), position = "stack")
```

- Try the other adjustments.

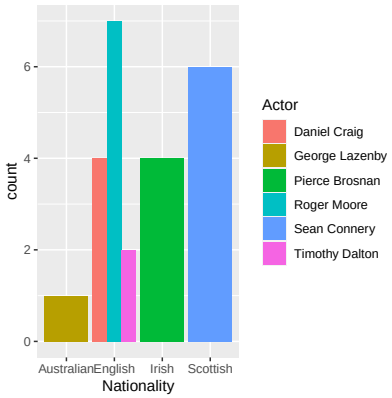

Figure 1: Barplot of ages using position = 'dodge'

Adjustment	Description
`dodge`	Adjust position by overlapping to the side
`fill`	Stack overlapping elements; standardise stack height
`identity`	Do nothing
`jitter`	Jitter points
`stack`	Stack overlapping elements

Table 1: Position adjustments - table 4.5 in the ggplot2 book.

Themes

Create **any** plot you like on either the `movies` or `bond` data set (or your own data if you like!)

Experiment with the different theme options using `+ theme_NAME()`. The in-built options are:

- `theme_gray()` – signature ggplot2 theme
- `theme_bw()` – dark on light ggplot2 theme
- `theme_linedraw()` – uses black lines on white backgrounds only
- `theme_light()` – similar to linedraw() but with grey lines aswell
- `theme_dark()` – lines on a dark background instead of light
- `theme_minimal()` – no background annotations, minimal feel.
- `theme_classic()` – theme with no grid lines.
- `theme_void()` – empty theme with no elements x For example:

```
ggplot(movies, aes(x = duration)) +
geom_density(aes(fill = classification), alpha = 0.5) + theme_bw()
```