

ggplot2 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.¹ 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`).

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

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

Facetting

3. Facetting is a mechanism by which we can recreate all layers in a plot, over another variable. For example we could take the boxplots from the previous question and recreate the same collection with a separate pane for each of the genders by adding a `facet_wrap(~ sex)` layer. Give it a try.
4. Facetting allows quick creation of visualisations that examine multiple properties of a data set. Take the final plot from the bar charts section above and facet on orientation

This graph tells lets us compare the proportions of males and females for each body type, amongst the bisexual, gay and straight populations on okcupid. Experiment with different columns for facetting, e.g. `diet`.