broom solutions

Jumping Rivers

Question 1

First, let's load the required packages and data

```
library("broom")
library("tidyverse")
library("jrTidyverse2")
library("GGally")
data(beer, package = "jrTidyverse2")
```

This data contains roughly 1500 beers, their alcohol percentage and their colour. Colour is ranked from 1-50, with 50 being the highest.

1) Use `?beer` and `head(beer)` to get a brief overview of the data.

2) We are going to look at how the colour of a beer affects the alcohol percentage using linear regression.

```
fit = lm(ABV ~ Color, data = beer)
```

The above code will run a linear regression model with alcohol percentage, `ABV`, as the response and colour, `Color`, as a variable. Explore the output of `summary(fit)`. Without using **broom**, what is the p-value for the variable `color`?
Hint: use `summary()`

```
summary(fit)$coefficients[,4]
```

3) That method of grabbing the p-values was tiresome wasn't it? Tidy and store the output of `fit()` such that it is easier to grab the p-values.

```
tidy_fit = tidy(fit)
tidy_fit$p.value
```

4) Using the **GGally** package, produce a coefficient plot.

```
ggcoef(fit, exclude_intercept = TRUE,
errorbar_height = 0.5, vline_color = "red")
```

5) Now we are interested in visualising how well the model has performed. We can do this using the fitted values. Store the data along side the fitted values from the model.
Hint: use `augment()`

```
aug_fit = augment(fit)
```

6) Amend the code in the notes given to make Figure 1.3 to plot the fitted values against the original data. Alternatively you can use base R to perform this task by using `plot()` and `points()`. Does it look like the model has performed adequately?

```
aug_fit %>%
  select(ABV, Color, .fitted) %>%
  gather(Type, Value, -Color) %>%
  ggplot(aes(x = Color)) +
  geom_point(aes(y = Value, colour = Type))

## OR
```

```
plot(aug_fit$Color, aug_fit$ABV)
points(aug_fit$Color, aug_fit$.fitted, col = "red", type = "l")
```

7) Adjusted R squared is a measure of how well the model is explaining the variation in the data. Technically, it is a measure of how close the original points are to the fitted values. It can take a value between 0 - 1. Lower would mean the model explains no variation and therefore is not very good whilst higher would mean the model explains all of the variation and therefore is very good. Using `glance()`, what is the adjusted R squared for the model? How good is the model at explaining the variation in the data?

Question 2

The functions within **broom** can be used on the outputs of lots of statistical functions in R, not just linear regression. To get a full list of the functions **broom** works on, go to the **broom** GitHub page and scroll to the bottom.

https://github.com/tidyverse/broom

To demonstrate this, we're going to look at another type of statistical inference, a t-test. The command for a t-test in R is `t.test()`.

```
data(movies, package = "ggplot2movies")
test = t.test(budget ~ Action, data = movies)
```

Here we are performing what is called a "two-sample" t-test where we are asking if the budget is the same for each category in `Action`. Basically, we are testing if the average budget is the same within each category of the variable `Action` i.e. non-action and action movies.

1) *Without* using **broom** or a t-test, work out the mean budget within each category of the variable `Action`.
Hint: Use **dplyr** and watch out for NAs!

```
movies %>%
  group_by(Action) %>%
  summarise(mean(budget, na.rm = TRUE))
```

2) Explore the output of the this t-test by calling the t-test object, `test`. Do the means in each group agree with yours?

```
test
```

3) Using **broom**, what is the confidence interval for the difference in means and what is the p-value for the test?

```
tidy(test) %>%
  select(conf.low, conf.high, p.value)
```

4) A p-value for this t-test is a measure of how sure we are there is a difference in means in each group. Here a p-value below 0.05 would indicate that we are sure there is a difference in means. Using the answers to 1) and 3), what can you conclude about the budget for action and non-action movies?

5) Harder: Why do you think `augment()` doesn't work for t-tests?

```
# Augment doesn't work on t-tests as there is no meaningful sense
# in which a hypothesis test produces output about each initial data point.
```