purrr practical

Jumping Rivers

First we must load the **tidyverse**

```
library("tidyverse")
```

Question 1

```
data(l, package = "jrTidyverse2")
```

a) `l` is a list with 3 vector elements; `A`, `B`, `C`. Each vector contains the heights of a different class at a university.

- How many students are in each class?
- Which class has the highest average height?
- Which class has the smallest student?
- Which class has the tallest student?
  Hint: use `map()`, `map_dbl()`, or `map_df()`

b) For each class, work out the lower and upper confidence interval bounds for the mean given that they are $\mu - 1.96\sigma$ and $\mu + 1.96\sigma$ respectively. Where μ is the sample mean and σ is the sample standard deviation.
Hint: use the formula notation and either `map()`, `map_dbl()` or `map_df()`.

Question 2

Now we're going to look at a list containing happiness rankings for countries around the globe.

```
data(happiness, package = "jrTidyverse2")
```

a) How long is the list? Is this a recursive list? How many countries does the list contain information on? For each country how many pieces of information is there?
Hint: use `str()`

b) Return the name of each country contained in the list. To make it a bit easier to read return the output as a character vector.

c) Try `names(happiness)`, what happens? Use the answer to `b)` to rename each element of the list after it's representative country.

d) What has the UKs average happiness rank been over the last 3 years?

e) Over the last 3 years, what is the average happiness score for every country? Store this in a vector of doubles.

f) Which region of the world has the high average happiness score?
Hint: store the region for each country in a vector, combine it into a data frame with the average happiness then use **dplyr**.

g) Using **ggplot2** and `geom_col()`, plot the answer to f) as a bar chart

Question 3

Load the data and the required packages

```r
library("broom")
data(beer_tidy, package = "jrTidyverse2")
```

Here we have a data set of around 500 beers along with their alcohol percentage, colour and type.

```r
head(beer_tidy)
```

We're going to run a linear regression on the data testing how colour affects alcohol percentage. However, using **purrr**, we are going to do it for each different type of beer.

1) Nest the beers within each type. Save this data in a variable called `beer_nest`.
2) For each type of beer, fit a linear regression model with alcohol perecentage as the response variable and colour as the predictor. Store the models in a list column next to your nested data. To help you out, the below code is how you would fit the model to the entire data set, before nesting.

```r
fit = lm(ABV ~ Color, data = beer_tidy)
```

Hint: use `mutate()` and `map()`

3) For each model, mutate a new column with the tidied models in. We want to plot the points eventually, so we want to tidy the model such that it returns the fitted points.
   Hint: use `mutate()`, `map()` and `augment()`
4) Select the columns containing the beer type and the tidied models and unnest the data.
5) Given you have stored unnested models in the variable `beer_nest()`, the following code WILL plot the lines made by the linear regression models

```r
ggplot(beer_nest) +
  geom_line(aes(x = Color, y = .fitted, colour = Type), size = 2)
```