Statistical Programming - Week 3

October 4, 2018

Question 0

Often when writing computer code, we wish to do something multiple times. In R, this is done by writing a 'for loop'. Type the following into R, and run it:

```
for (i in 1:5) {
  print("hello")
}
```

The output should be:

```
[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"
```

This code tells R to print 'hello' 5 times. How does it work? Remember that in R, 1:5 gets turned into the vector (1,2,3,4,5):

```
> 1:5
[1] 1 2 3 4 5
```

So the first line is essentially ?for (i in (1,2,3,4,5))?. R runs through the body of the loop once for each number in this vector. The body simply tells R to print 'hello'.

Here is a slightly more complex loop:

```
for (i in 1:5) {
  print(i)
}
```

The output is:

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

As before, the body of the loop gets executed 5 times. The first time through, the variable 'i' is assigned the value '1', which is the first element of the vector (1,2,3,4,5). The second time the body is executed, 'i' gets assigned the value 2, and so on.

So 'i' changes value each time the body is executed.

Exercise: Write a for loop that prints out all the even numbers between 2 and 100. (Hint: use the 'seq' command).

Question 1

We will explore the accuracy of bootstrap confidence intervals for estimating the unknown mean μ of a Normal distribution, with known $\sigma = 1$. We can generate sample data using the **rnorm** function in R. We will assume that the (unknown) mean is equal to 2. Suppose the sample size is $n = 5$. Generate some data using:

```
set.seed(1)
y <- rnorm(5,2,1)
```

(note: the set.seed() command initialises the pseudo-random number. generator in R to a fixed point. This means that. you will get the. same 'random' numbers every time you run this code. Check that your value of y is equal to 1.373546, 2.183643, 1.164371, 3.595281, 2.329508)

Find a 95% confidence interval for the unknown mean using bootstrap, and compare this to the exact confidence interval (using qnorm). How does the accuracy increase when you increase the sample size and generate data with $n = 50$ and $n = 200$?

Question 2

A gambler is interested in determining whether a coin is fair, in the sense of coming up heads with a 0.5 probability. Let p denote the probability of the coin coming up heads. The coin is tossed 100 times, with 60 of the tosses landing heads.

1. Compute a sensible point-estimate for p based on this data.
2. Compute a 95% confidence interval for p using bootstrapping. Hint: to get this data into R, you can use the rep() function: $y \leftarrow c(rep(0, 40), rep(1, 60))$
3. Now suppose that there were 1000 tosses, with 600 heads obtained. Compute a point estimate for p, and a 95% confidence interval. Interpret your results, compared to the case with 100 tosses.

Question 3

A researcher wishes to compare the heights of people in two UK villages. They have collected a random sample of 10 people from each village. The data for the two villages are:

```
y1 <- c(153.7, 161.8, 151.6, 176.0, 163.3, 151.8, 164.9, 167.4, 165.8, 156.9)
y2 <- c(165.1, 153.9, 143.8, 127.9, 161.2, 149.6, 149.8, 159.4, 158.2, 155.9)
```

The goal is to test whether there is a significant different in heights between the two populations. If the data were known to be Normal, this could be. done using a Student-t test, i.e:

```
t.test(y1,y2)
```

However, there is no prior reason to believe the data sets come from a Normal distribution so the researchers do not want to make this assumption. Instead, they will use a nonparametric test based around bootstrapping.

Use a bootstrap to find a confidence interval for the difference between the two population means. Recall that by the duality of confidence intervals and hypothesis testing, rejecting the null hypothesis of 'no difference in means' at the $\alpha = 0.05$ level is equivalent to the 95% confidence interval for the difference in means not containing 0. Hence decide whether the difference in means is significant at the 0.05 level

Question 4

We will use the bootstrap to find confidence intervals for the coefficients in a linear regression.

CD4 cells are carried in the blood as part of the human immune system. One of the effects of the HIV virus is that these cells die. In a study of the effectiveness of a new anti-viral drug on HIV, 20 HIV-positive patients had their CD4 counts recorded and then were put on a course of treatment with this drug. After using the drug for one year, their CD4 counts were again recorded.

$x_i(baseline)$: The CD4 counts (in 100?s) on admission to the trial.

$y_i(oneyear)$: The CD4 counts (in 100?s) after one year of treatment with the new drug.

This data is contained in the 'boot' package. Try to load this package by typing:

```
library(boot)
```

If this works, then great. However you might get an error saying the package is not installed. In this case, you can install the package by typing:

```
install.packages('boot')
library(boot)
```

Note that the 'install.packages' command only ever needs to be run once; the package will then be installed on your computer indefinitely. But you need to type library(boot) whenever you restart R in order to load the package.

After the package is loaded, the data is stored in the **cd4** variable. You can view it by typing 'cd4' We consider the following linear regression model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

Our aim is to fit the model and find confidence intervals for β_0 and β_1 without making specific distributional assumptions about the error distribution (i.e. without assuming ϵ has a Normal distribution). It is easy to obtain estimates about the regression parameters. Recall that the least squares method is a non-parametric fitting technique. The least squares fit

```
mod <- lm(oneyear~baseline,data=cd4)
summary(mod)
```

to this data gives point estimates $\beta_0 = 0.69$ and $\beta_1 = 1.03$. Standard errors and CIs for the parameters, however are typically obtained based on asymptotic normality arguments. On the other hand bootstrap methods can be used for this purpose as a non-parametric alternative.

Use a non-parametric bootstrap to re-sample from the rows of the data matrix, i.e., re-sample pairs (x_i, y_i), re-fit the model to each bootstrap sample, form the bootstrap distribution of the βs, and hence find their confidence intervals. Hint: To resample from the rows of the data matrix first draw i from a discrete uniformon [1, . . . , n] and then extract (x_i, y_i):

```
n <- dim(cd4)[1] # sample size
index <- 1:n # patient sample number
index_star <- sample(index, size=n, replace=TRUE) # draw n discrete uniforms
cd4_star <- cd4[index_star, ] # obtain boostrap sample
```