Solution to Lab 4

Univariate Statistics with R

Some notes and observations from the labs so far

Everyone has made a lot of progress so far and it is great for us to see students engaging with R and hopefully seeing the utility of using this tool. There are a few points worth reiterating here to clear up any possible confusion and to help you work better in the remaining labs.

Work using scripts

If you recall the first lab notes you were instructed to create a new project within R-Studio for the lab (refer back to these notes if you have forgotten). By creating a project for each lab, or each piece of work, you create a home for all files relating to the project, which can make your workflow much smoother.

More important though is the use of scripts. If you press *ctrl+shift+n (or cmd+shift+n)* you will create a new empty text file in the Editor section of R-Studio. You can then write your commands here and *send* them to the console to be executed in several ways. You can highlight whole blocks of code to be sent at once, or you can send one particular line. To send a block, highlight the commands you wish to run, and press *ctrl+Enter (or cmd+Enter)*. To send a single line, simply put the cursor somewhere on the line you wish to send and hit the same keyboard shortcut.

This is a really important point. Working using a script is useful for many reasons. It allows you to easily go back through the code you have written and spot any mistakes - things that might not initially throw an error but are problematic down the line. For some of you this happened last week in terms of accidentally adding extra variables to your dataframes etc. Also, it allows you to be able to share your work easily with other people. Perhaps most importantly, given the early stage you are at in your R life, it allows you to look back over your previous labs code and remind yourself how to complete certain tasks. This leads nicely on to the second major observation...

Comment your code extensively

Look at the following examples of code blocks:

```
choose <- c("football", "running", "hockey", "golf", "swimming")
sport <- sample(choose, 100, replace = TRUE, prob = c(.1, .5, .2, .1, .1))
data <- cbind(data, sport)
```

```
# First we create a list of the names of our sports
choose <- c("football", "running", "hockey", "golf", "swimming")
# next we create a variable by sampling from this list 100 times with replacement
sport <- sample(choose, 100, replace = TRUE, prob = c(.1, .5, .2, .1, .1))
# add the sport vector as a new column to our 'data' dataframe
data <- cbind(data, sport)
```

The # character is used to indicate that a line of text is a comment and should be ignored by R. You can use it to describe what you are doing and still be able to highlight the whole block and run the code (as R will ignore the comment lines). It is ALWAYS good to comment your code, even if you don't think so at the time but it is especially useful to beginners as it allows you to explain in words what commands are doing. This encourages you to work through in your head what some R commands do which will help your understanding

more than just copying the code into the console. Additionally, code you write today might make sense to you right now but that may not be the case several months later when you come back to it.

Commenting your scripts will help you learn to use R and it will make sharing code with other people easier.

Dollar Notation $

You might have picked up on different methods we used to access specific variables in the dataframes we created.

```
head(mtcars) # to see the top 5 ros of the mtcars dataset
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
#Two different ways to specifically pick the first 5 values of the mpg variable:
mtcars[1:5, 1]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7
```

```
mtcars$mpg[1:5]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7
```

As with most tasks we have to complete in R, there are generally many ways to get the desired end result. For instance, the first variable in the *mtcars* dataset is *mpg* and so we can access the values of the mpg variable by asking for the first column. We can do this using the square brackets **[r,c]** where we replace *r* with the rows we want and *c* with the columns we want. As *mtcars* is a dataframe, we can refer to the variable by name by using dollar notation: **dataframe$variable**. This method makes code more readable and also avoids pitfalls such as broken code after columns are re-ordered. It's important that you are aware of both ways as dollar notation is specific to dataframes and there are times you want to select values out of other data types. However in general, you will be working with dataframes and when that is the case, dollar notation is your friend.

Welcome to lab 3

Last weeks lab was a big one and provided a lot of code for you to execute and evaluate what was happening. The purpose was to get you familiar with working in the R environment. It can seem intimidating to those with no previous programming knowledge but hopefully the more you write and execute commands and see the results, the more comfortable you will feel with R. The focus of this weeks lab is a bit more on working out problems. It may be useful to work in small groups for this lab. Of course, there is a little bit of plotting as well - we don't want to let those newly aquired skills fade. If you finish the exercises in this worksheetand and are still keen on more R, there is an additional document for this lab that introduces you to the *ggplot2* package.

Some basic probability

Task 1: The equation for conditional probability is:

$$p(B|A) = \frac{p(A \bigcap B)}{p(A)}$$

In words, what is this equation saying?

ANSWER: the probability of B given A is equal to the probability of A and B over the probability of A.

Task 2: I have a bag with 40 marbles of different colours. There are 10 red marbles, 7 blue, 15 yellow and 8 green. If I do a simple experiment where a draw a single marble randomly with replacement, what is:

1. p(green) **ANSWER: 8/40 = 0.2 or 1/5**
2. p(red or yellow) **ANSWER: (10 + 15)/40 = 0.625 or approx 6/10**
3. p(not blue) **ANSWER: (40 − 7)/40 = 0.825 or approx 8/10**
4. p(colourless) **ANSWER: 0/40 = 0**

Task 3: I now repeat the experiment above, but this time I sample without replacement. What is the probability of the following sequences;

1. First red, second red, third blue. **ANSWER: 10/40 * 9/39 * 7/38 = 0.010 or approx 1/100**
2. First blue, second blue, third blue. **ANSWER: 7/40 * 6/39 * 5/38 = 0.0035 or approx 4/1000 or 1/250**

Hint: Remember if we are keeping this simple with respect to independence, we can still use the products of the probabilities.

Samples

Here is a question which does not require R, but does require you to think carefully. Be sure to talk this out with a classmate.

Task 6: Think about random samples. In the lecture we largely assumed independent random samples for our simple experiments. Navarro makes the point that in practical terms, we do not necessarily need this, but instead just need to make sure that we have randomly selected on the variables we are interested in. So, think about your area of psychology, think about the most common variables of interest, and think about the most common sample compositions. Could these be considered random in the probability or Navarro sense? And why?

ANSWER: This is an open discussion. No set answer. But an example may be researching IQ in a student sample where we may expect being at university means you are above average on IQ in the first place

Tabulating Joint Events

Task 7: In a university department, there are 50 lecturers, 29 male and 21 female. The lecturers were asked if they would like a new coffee machine, 40 said yes, 10 said no (the fools!). Create a 2x2 table for this joint event, and calculate the cell probabilities.

ANSWER: tables that look something like this

	Yes	No	Marginal
Male	p=.46	p=.12	0.58
Female	p=.34	p=.08	0.42
Marginal	0.80	0.20	1.00

Task 8: A tricky one! Can you think how we would go from these cell probabilities to expected frequencies? (We have not discussed this in class - yet!)

ANSWER: multiply by N

Task 9: Write a R script to build a table with a structure like that in the lecture slide. (and below; ***Note***: The numbers will be different for the numbers above - this is just here to show no. of rows, columns and labels etc.)

	Yes	No	Marginal
Male	p=.10	p=.45	0.55
Female	p=.30	p=.15	0.45
Marginal	0.40	0.60	1.00

You can use the commands from last weeks lab to help you (**Hint**: look at `matrix()`, `rownames()` and `colnames()`; and think about indexing)

```
# this is clunky but works
empty <- matrix(nrow = 3, ncol = 3)
colnames(empty) <- c("Yes", "No", "Marginal")
rownames(empty) <- c("Male", "Female", "Marginal")
empty[1,1] = .46
empty[1,2] = .12
empty[1,3] = .58
empty[2,1] = .34
empty[2,2] = .08
empty[2,3] = .42
empty[3,1] = .80
empty[3,2] = .20
empty[3,3] = 1.00
empty
```

```
##           Yes   No Marginal
## Male     0.46 0.12     0.58
## Female   0.34 0.08     0.42
## Marginal 0.80 0.20     1.00
```

```
# here is a much nicer way (thanks to Martin C)
# This solution is not only quicker to plot
# but will calculate the cell probabilities
# by calculating outer product of the marginal vectors

# create marginals
marg_sex = c(29, 21)/50
marg_cof = c(40, 10)/50

# calculate outer products
# type ?`%o%` (backtick ` is to the left of the 1 key)
# for help file to learn more about the %o% operator
prod = marg_sex %o% marg_cof

# Add marginals
prod = rbind(prod, marg_cof)
prod = cbind(prod, c(marg_sex, 1.00))

# add labels to columns
```

```r
colnames(prod) <- c("Yes", "No", "Marginal")
rownames(prod) <- c("Male", "Female", "Marginal")
prod
```

```
##            Yes    No Marginal
## Male     0.464 0.116     0.58
## Female   0.336 0.084     0.42
## Marginal 0.800 0.200     1.00
```

The Binomial in R

The title of this section should give a clue as to which set of functions you need to be using to answer these questions.

N is the number of flips, think of the number of 0s as the number of ***HEADs*** *and the 1s represent* ***TAILS***

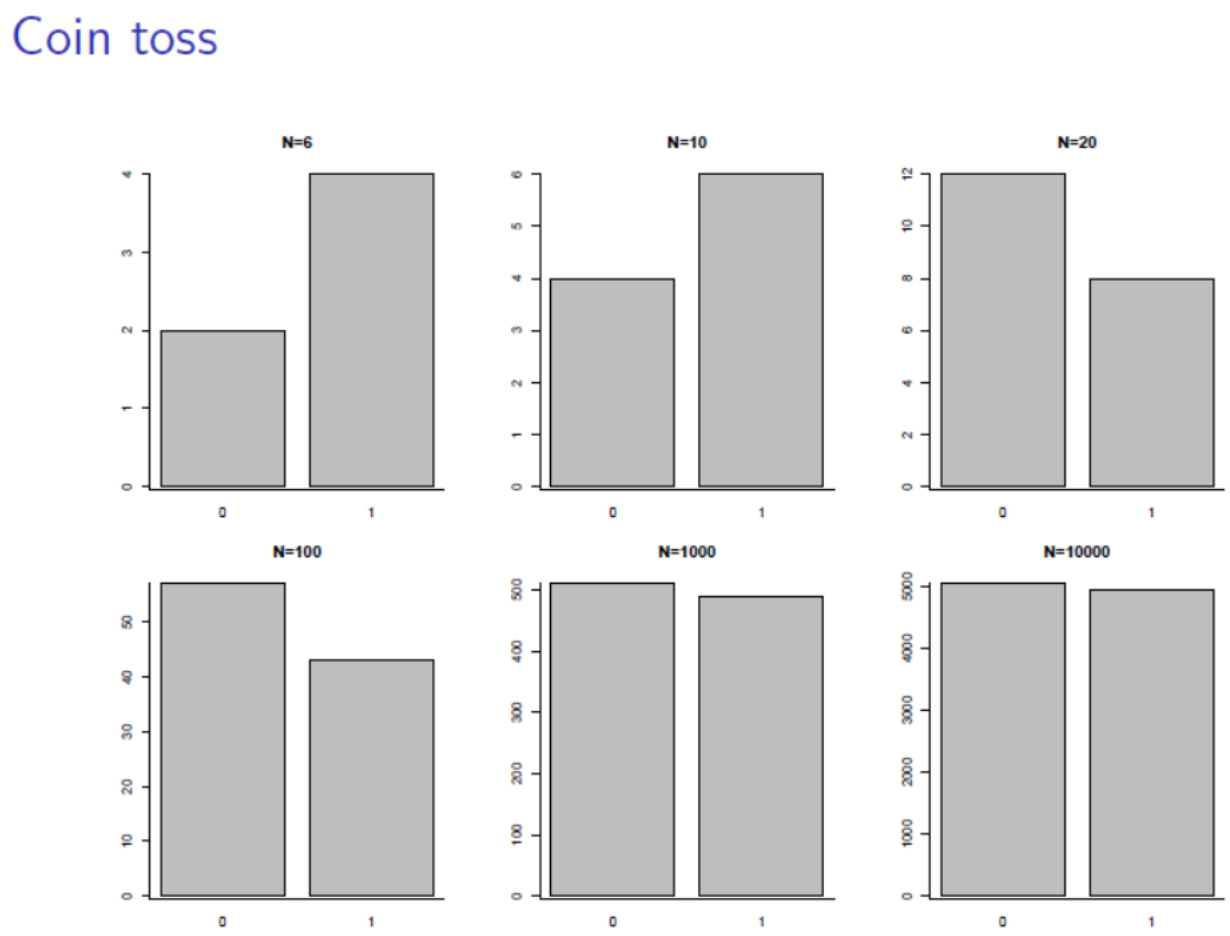
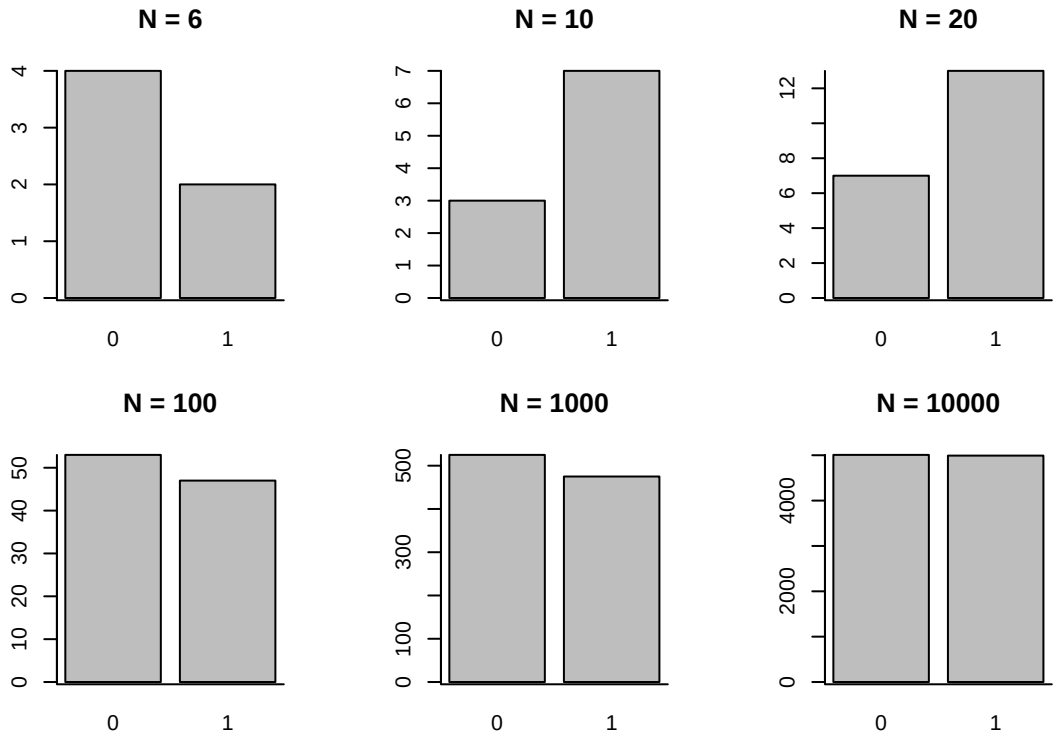


Figure 1: Coin Toss Plots

Task 10: Replicate the plots in coin toss plots (reproduced above). Create a single figure of the same format. (**Hint**: Look back at the last sections of last weeks lab for how to combine multiple plots into one figure, and also see `par()`, `mfrow()` and `mar()`)

```
### ANSWER ###

# code used in slides is here
par(
  mfrow = c(2,3),
  mar = c(2, 4.1, 4.1, 2.1))
plot(as.factor(rbinom(6, 1, .50)), main = "N = 6")
box(bty = "L")
plot(as.factor(rbinom(10, 1, .50)), main = "N = 10")
box(bty = "L")
plot(as.factor(rbinom(20, 1, .50)), main = "N = 20")
box(bty = "L")
plot(as.factor(rbinom(100, 1, .50)), main = "N = 100")
box(bty = "L")
plot(as.factor(rbinom(1000, 1, .50)), main = "N = 1000")
box(bty = "L")
plot(as.factor(rbinom(10000, 1, .50)), main = "N = 10000")
box(bty = "L")
```




```
par(
  mfrow = c(1,1),
  mar = c(5.1,4.1,4.1,2.1))
```

How do your results look when compared to those above?

ANSWER: general idea here is the first row could look very different as it is a small number of trials. The latter plots should look the same

Task 11: For this question, look at Navarro pages 283-288 (Version 0.5) for the information we can get from `rbinom`, `pbinom`, `qbinom` and `dbinom`.

I have set a stats test that has 15 multiple choice items. Each item has 4 possible answers. If students simply guess randomly for each question:

1. What is the probability that students will get **exactly** 6 answers correct?

```
### ANSWER ###

dbinom(x = 6, size = 15, prob = 1/4)
```

```
## [1] 0.09174777
```

2. What is the probability of a student getting 6 or fewer answers correct?

```
### ANSWER ###

pbinom(q = 6, size = 15, prob = 1/4)
```

```
## [1] 0.9433797
```

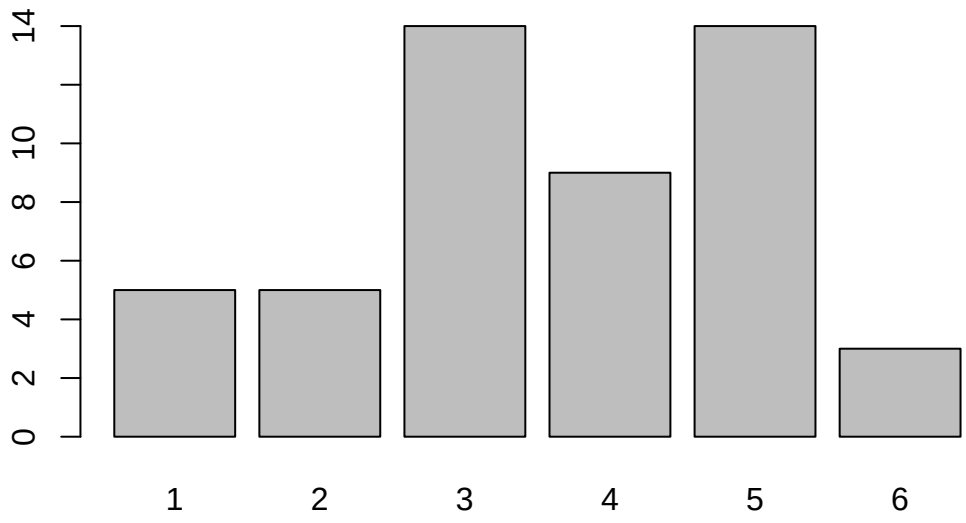
3. What score should we expect 50% of the class to achieve?

```
### ANSWER ###

qbinom(p = .50, size = 15, prob = 1/4)
```

```
## [1] 4
```

4. Let's assume I have given this test to the last 50 classes, generate some data and plot it.

```
### ANSWER ###

plot(as.factor(rbinom(n = 50, size = 15, prob = 1/4)))
```



That's it for the 'stats stuff' this week. If you are hungry for more, there is an extra set of materials this week that presents an introduction to ggplot2 along with some exercises. It introduces an alternative graphics plotting system to the base plot() we have been using. It is a very popular way of plotting data and many find it more intuitive - and prettier - than plot(). The exercises will also help you gain some more fmailiarity with R and the flexibility it provides.