**Introduction to R and RStudio  
Post-session activities**

1. **Creating vectors**
   1. Create a vector containing all the integers from 1 to 100 (inclusive) and store it as an object called x
   2. Create a new vector containing all the values of x raised to the power of ½ and store it as an object called y   
      **Hint:** fractional indices need to be wrapped in brackets, 5^3/4 is not the same as 5^(3/4)
   3. Print out a logical vector indicating whether each element in y is equal the square root of its corresponding value in x  
      **Hint:** use the == operator to compare two sets of values
   4. What does the result of (3) tell you about a number raised to the power ½?
2. **Creating lists and subsetting**
   1. Create a list and store it as an object called my\_list, containing the following data:
      1. a character vector with the values “blue”, “green”, “pink”
      2. a Boolean vector with the values TRUE, TRUE, FALSE
      3. a numeric vector with the values 7, 8, 9
   2. Extract the second element, of the 3rd element of the list  
      **Hint:** remember that to get the nth element of a list we use [[n]] and to get the nth element of a vector we use [n]
3. **Using functions**
   1. The runif() function randomly draws a value from a uniform distribution. Use the ? operator to open the help page for the runif() function
   2. Use the runif() function to generate 10 random values between 0 and 1. Did you need to specify the min and max arguments? Why?
   3. Create your own function called percent that takes two mandatory arguments, *a* and *b*, and returns a as a percentage of b (e.g. percent(3, 4) should return 75)
   4. Overwrite your function to replace it with one that includes an optional argument *round* with a default value of FALSE. When round is FALSE, the function should return the same values as before, but when round is TRUE, the function should return the value rounded to the nearest integer  
      **Hint:** use if(<condition>) {<do something if TRUE} else {<do something if FALSE}, and use the round() function to round the values
   5. Compare the value of percent(3, 14, round = FALSE) to percent(3, 14, round = TRUE)
4. **Working with data.frames**
   1. Read in the winemag-data\_first150k.csv file and store it as a data.frame called wine
   2. Use the str() function to get an understanding of the structure of the data. How many rows and columns are there?
   3. Extract the column price using the $ operator, and subset it to remove any wines more expensive than $200 (because who spends that much on wine?!) and store is as an object called price\_subset
   4. Use the hist() function to plot a histogram of the distribution of wine prices. Experiment with different values for the *breaks* argument.
   5. Load the dplyr package with library()
   6. Use the dplyr function filter() to filter for Sauvignon Blanc wines cheaper than less than or equal to $20. Arrange this data in order of the *points* variable (highest to lowest) and store it in an object called sauv\_blanc  
      **Hint:** use arrange(<data>, -points) to arrange in descending order of points
   7. Extract the description of the highest rated wine in sauv\_blanc  
      **Hint:** if you arranged sauv\_blac correctly, the highest rated wine is the first one, so you could use [<row index>, <column index/name>] to subset the row and column you want.
   8. Pipe together the following function calls:
      1. use group\_by() to group the wine dataset by country and province
      2. use the summarise() function to calculate a variable called *score*, that is the median of the points variable
      3. use the arrange() function to arrange the data in order of the score variable from highest to lowest
      4. What is the highest rated country for wine production? Controversial?!
   9. Use the mutate() function to create a new column called points\_per\_price that is the points variable divided by the price variable

**If you have any questions or want feedback, please contact** [**hefin.rhys@ucb.com**](mailto:hefin.rhys@ucb.com)