R Crash Course - Subsetting and logical tests

Willson Gaul
willson.gaul@ucdconnect.ie

January 2021

Introduction

When working in R, you must break large tasks into smaller steps, and then write code to do each of those steps. Remember, a computer does one thing at a time, and it does them in order. Before you write a line of code, write a comment in normal language saying what you want to do. If you can't say what you want to do in English, you definitely won't be able to say it in R.

This document starts by giving you each line of code that I ran during the demonstration at the start of the workshop. You should write each of the lines of code yourself into your own script, run them, and make sure you understand what they do. Then, starting on step 18 are new tasks but no code to complete them. However, this document *does* show some output showing what it should look like if you do the task correctly. You should write comments and code to do these tasks, and make sure your results match the results shown in this document.

Key Points

- write comments for every line
- break tasks into small steps
- take a look at the data after every step to make sure it did what you want.

Before you start these tasks, you should create a new folder on your computer. You can name the folder anything you want. You will save all the files for these practice tasks in your new folder.

Practice tasks

1) Download and save HEIGHT.CSV (the file is attached to the email I sent you)

2) Make and save a new R script.

3) Set the working directory (`setwd()` or Session >> Set Working Directory >> Choose Directory).

```
# set working directory (your file path will be different from mine)
setwd("~/Documents/UCD/demonstrating/R_crash_course/practice_tasks/")
```

4) Make a vector of numbers, and call the vector `nums`. (From now on you should be working in your R script, and writing comments).

```
# make a vector of numbers, and name it nums
# Remember the <- and the = do the same thing so you could also
# write this line of code as
# nums = c(1, 5, 2, 4, 7)
nums <- c(1, 5, 2, 4, 7)
```

5) Take a quick look to make sure this did what you want.

```r
# print nums to the screen to make sure this did what I want
nums
```

```
## [1] 1 5 2 4 7
```

6) Get the 2^{nd} value from nums. What do you expect to get?

```r
# get the 2nd value from nums.  I expect to get a 5 printed to the screen.
nums[2] # square brackets are used for subsetting.  The 2 gives me the 2nd element.
```

```
## [1] 5
```

7) Get the 3^{rd} and 5^{th} values

```r
# Get the 3rd and 5th values
nums[c(3, 5)]
```

```
## [1] 2 7
```

8) Get all values from `nums` that are bigger than 3

```r
# get values bigger than 3 from nums
nums[nums > 3]
```

```
## [1] 5 4 7
```

9) Just to demonstrate, lets look at that previous task in more detail, looking at what is happening in the code. Make sure you understand why each line of code gave that output. Compare the code and the output to `nums` to make sure it is doing what you expect. To demonstrate:

```r
# print nums to the screen so I remember what is in it
nums
```

```
## [1] 1 5 2 4 7
```

```r
# this will print a bunch of TRUE / FALSE values, one for each element of
# nums, showing whether that element is greater than 3
nums > 3
```

```
## [1] FALSE  TRUE FALSE  TRUE  TRUE
```

```r
# I can subset nums using a vector of TRUE / FALSE values
nums[c(TRUE, FALSE, TRUE, TRUE, FALSE)]
```

```
## [1] 1 2 4
```

```r
# change one TRUE / FALSE value from above to see what happens.  I will change
# the first value from TRUE to FALSE
nums[c(FALSE, FALSE, TRUE, TRUE, FALSE)]
```

```
## [1] 2 4
```

```
# instead of writing the TRUE / FALSE values out myself, I can put a logical
# test inside the square brackets.  The logical test makes a vector of
# TRUE/FALSE values that are used to subset nums.
nums[nums > 3]
```

```
## [1] 5 4 7
```

10) Read in the data from HEIGHT.CSV

```
# read in the height data
height <- read.csv("HEIGHT.CSV")
```

11) take a look to make sure the data look ok

```
# look at the first few rows of the height data
head(height)
```

```
##   AGE SEX YEAR.MEASURED GLASSES HANDED RANK HEIGHT WEIGHT
## 1  20   M            87 GLASSES      2    1  1.841   76.9
## 2  20   M            87 GLASSES      1    1  1.748   84.4
## 3  20   M            87 NEITHER      2    1  1.647   72.0
## 4  20   M            87 NEITHER      1    1  1.761   83.0
## 5  22   M            87 NEITHER      1    1  1.749   68.4
## 6  29   M            87 NEITHER      1    2  1.739   92.8
```

```
# look at the number of rows and columns, and the data types of each column
str(height)
```

```
## 'data.frame':    2298 obs. of  8 variables:
##  $ AGE          : int  20 20 20 20 22 29 20 20 21 21 ...
##  $ SEX          : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ YEAR.MEASURED: int  87 87 87 87 87 87 87 87 87 87 ...
##  $ GLASSES      : Factor w/ 4 levels "BOTH","CONTACTS",..: 3 3 4 4 4 4 4 3 3 4 ...
##  $ HANDED       : int  2 1 2 1 1 1 1 2 1 1 ...
##  $ RANK         : int  1 1 1 1 1 2 1 1 1 1 ...
##  $ HEIGHT       : num  1.84 1.75 1.65 1.76 1.75 ...
##  $ WEIGHT       : num  76.9 84.4 72 83 68.4 92.8 77.4 98.5 86.4 67.4 ...
```

```
# look at some summary statistics to make sure the values are reasonable
# (i.e. no negative or super big height values)
summary(height)
```

```
##       AGE          SEX      YEAR.MEASURED       GLASSES         HANDED     
##  Min.   :20.00   F:1364   Min.   :87.00   BOTH    : 241   Min.   :1.000  
##  1st Qu.:22.00   M: 934   1st Qu.:87.00   CONTACTS:   90   1st Qu.:1.000  
##  Median :24.00            Median :88.00   GLASSES : 580   Median :1.000  
##  Mean   :23.92            Mean   :87.61   NEITHER :1384   Mean   :1.115  
##  3rd Qu.:26.00            3rd Qu.:88.00   NA's    :   3   3rd Qu.:1.000  
##  Max.   :29.00            Max.   :88.00                   Max.   :2.000  
##                                                           NA's   :4      
##       RANK           HEIGHT          WEIGHT     
##  Min.   :1.000   Min.   :1.428   Min.   : 41.30  
##  1st Qu.:1.000   1st Qu.:1.613   1st Qu.: 59.20  
##  Median :1.000   Median :1.680   Median : 66.40  
##  Mean   :1.615   Mean   :1.685   Mean   : 68.28  
##  3rd Qu.:2.000   3rd Qu.:1.749   3rd Qu.: 76.20  
##  Max.   :4.000   Max.   :2.042   Max.   :124.30  
##  NA's   :5
```

12) Get the 1^{st} row of the `height` data frame

Remember, when subsetting a data frame using square brackets, the first position indicates the rows, then there is a comma, and the second position indicates the columns (`my_data[rows, columns]`). Leaving a position blank (or with a space) gives ALL the elements.

```r
# get the 1st row (and all columns) of the height data frame.
height[1, ] # leave a space after the comma to get all columns
```

```
##   AGE SEX YEAR.MEASURED GLASSES HANDED RANK HEIGHT WEIGHT
## 1  20   M            87 GLASSES      2    1  1.841   76.9
```

13) Get the 2^{nd} column (all rows)

```r
# get the 2nd column from the height data frame
height[ , 2] # leave a blank space before the comma to get all rows
```

```
##    [1] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##   [35] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##   [69] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [103] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [137] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [171] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [205] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [239] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [273] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [307] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [341] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [375] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [409] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [443] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [477] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [511] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [545] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [579] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [613] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [647] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [681] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [715] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [749] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [783] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [817] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [851] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [885] M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M M
##  [919] M M M M M M M M M M M M M M M M F F F F F F F F F F F F F F F F F F
##  [953] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
##  [987] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1021] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1055] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1089] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1123] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1157] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1191] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1225] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1259] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1293] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1327] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1361] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1395] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1429] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1463] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1497] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1531] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1565] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1599] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1633] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1667] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1701] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1735] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1769] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1803] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1837] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1871] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1905] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1939] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [1973] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2007] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2041] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2075] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2109] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2143] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2177] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2211] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2245] F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
## [2279] F F F F F F F F F F F F F F F F F F F F
## Levels: F M
```

Yikes! That's a lot of F and M values. But that seems right, my data frame should have 2298 rows (based on step 11 above), so when I ask for all of the 2^{nd} column, I should expect 2298 F and M values, which is what I got.

14) Get rows 3 and 4, all columns

```
# get rows 3 and 4, all columns
height[c(3, 4), ]
```

```
##   AGE SEX YEAR.MEASURED GLASSES HANDED RANK HEIGHT WEIGHT
## 3  20   M            87 NEITHER      2    1  1.647     72
## 4  20   M            87 NEITHER      1    1  1.761     83
```

15) Get rows 3 and 4, columns 1 and 6

```
# get rows 3 and 4, columns 1 and 6
height[c(3, 4), c(1, 6)]
```

```
##   AGE RANK
## 3  20    1
## 4  20    1
```

16) Get data only for females, and save it as a new object

```
# get data for females and save it as a new object
#
# I am going to write that comment again, but closer to "computer talk" by
# spelling the column name and the value representing females precisely:
#
# get rows for which the SEX column has a value of F
f_heights <- height[height$SEX == "F", ]
```

17) Take a look to make sure this did what I want

```
# look at the first few rows
head(f_heights)
```

```
##     AGE SEX YEAR.MEASURED GLASSES HANDED RANK HEIGHT WEIGHT
## 935  29   F            87 GLASSES      1    2  1.836   74.5
## 936  20   F            87 GLASSES      1    1  1.524   54.8
## 937  23   F            87 NEITHER      1    1  1.836   80.6
## 938  22   F            87 NEITHER      1    2  1.662   60.3
## 939  24   F            87 GLASSES      1    1  1.621   53.8
## 940  27   F            87    BOTH      2    4  1.648   51.8
```

```
# Check to make sure there are fewer rows (because I expect to no longer have
# rows for males)
str(f_heights)
```

```
## 'data.frame':    1364 obs. of  8 variables:
##  $ AGE          : int  29 20 23 22 24 27 26 28 24 25 ...
##  $ SEX          : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
##  $ YEAR.MEASURED: int  87 87 87 87 87 87 87 87 87 87 ...
##  $ GLASSES      : Factor w/ 4 levels "BOTH","CONTACTS",..: 3 3 4 4 3 1 4 1 4 2 ...
##  $ HANDED       : int  1 1 1 1 1 2 1 1 1 1 ...
##  $ RANK         : int  2 1 1 2 1 4 4 4 4 4 ...
##  $ HEIGHT       : num  1.84 1.52 1.84 1.66 1.62 ...
##  $ WEIGHT       : num  74.5 54.8 80.6 60.3 53.8 51.8 60.9 58.5 84.8 54.3 ...
```

```
# look at summary statistics.  I should see no "M" values for the SEX column.
summary(f_heights)
```

```
##       AGE          SEX      YEAR.MEASURED       GLASSES          HANDED
##  Min.   :20.00   F:1364   Min.   :87.00   BOTH    :195   Min.   :1.000
##  1st Qu.:22.00   M:   0   1st Qu.:87.00   CONTACTS: 80   1st Qu.:1.000
##  Median :24.00            Median :88.00   GLASSES :381   Median :1.000
##  Mean   :24.09           Mean   :87.66   NEITHER :706   Mean   :1.105  
##  3rd Qu.:26.00           3rd Qu.:88.00   NA's    :  2   3rd Qu.:1.000  
##  Max.   :29.00           Max.   :88.00                  Max.   :2.000  
##                                                         NA's   :3      
##       RANK           HEIGHT          WEIGHT     
##  Min.   :1.000   Min.   :1.428   Min.   :41.30  
##  1st Qu.:1.000   1st Qu.:1.588   1st Qu.:55.90  
##  Median :1.000   Median :1.630   Median :60.95  
##  Mean   :1.652   Mean   :1.632   Mean   :61.41  
##  3rd Qu.:2.000   3rd Qu.:1.675   3rd Qu.:66.33  
##  Max.   :4.000   Max.   :1.870   Max.   :90.90  
##  NA's   :1
```

18) Get data for males who wear contacts (but do not wear glasses).

I will not give you the code for this. You should write the code yourself, and then make sure your results match what is shown below. To start, break this up into two steps:

a) get data only for males, and save it as a new object. This should give you a data frame with 934 rows and 8 columns (check to make sure that is what you got).
b) subset the males-only data to get only data for males who wear contacts. This should give you a data frame with 11 rows and 8 columns. The mean age of males who wear contacts should be 23.2, the mean height should be 1.7793, and the mean weight should be 81.5. (Hint: if you use the `mean()` function to calculate the mean, make sure to use the argument `na.rm = TRUE` to exclude missing (NA) values. So the code would look something like this: `mean(male_ht$HEIGHT, na.rm = TRUE)`).

19) Find the mean height of females who wear neither glasses nor contacts.

Break this up into small steps, and write comments describing what you are doing in each step.

The mean height of females who wear neither glasses nor contacts should be 1.6284405.

20) How many males aged 26 or older wear glasses but not contacts?

There 57 males aged 26 or older who wear glasses but not contacts.

Summary & Important Points

- Work in small steps. Save intermediate results as objects.
- Make tests that will give you TRUE/FALSE vectors, and use those vectors to subset your data.
- When using square brackets to subset data frames, use code similar to: `my_data[rows, columns]` (you can also subset data using the `subset()` function)
- Leaving a blank space in the above syntax gives you *all* rows or columns.