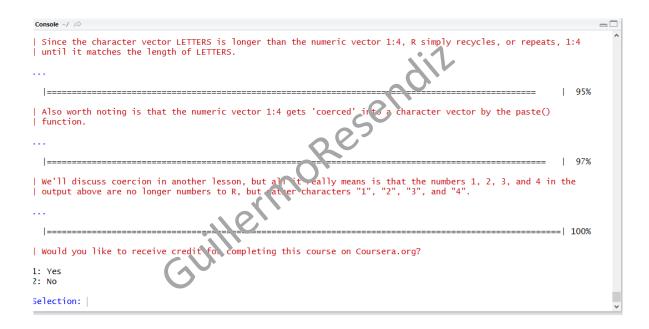
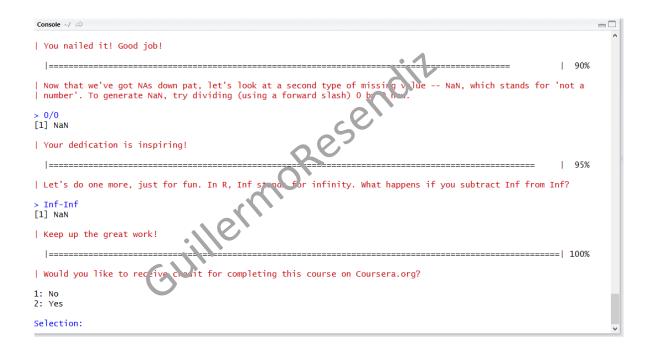
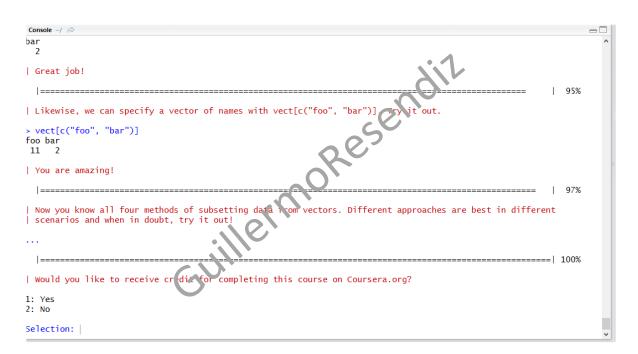
| Console ~/ ⇔ | |
|--|--------|
| You are amazing! | ^ |
| | = |
| Finally, let's pretend you'd like to view the contents of a variable that you created earlier, but you can't seem to remember if you named it my_div or myDiv. You could try both and see what works, or | |
| ======= 97% | = |
| You can type the first two letters of the variable name, then hit the Tab key (possibly more than once). Most programming environments will provide a list of variables that you've created that begin with 'my'. This is called auto-completion and can be quite handy when you have many variables in your workspace. Give | Ė |
| it a try. (If auto-completion doesn't work for you, just type my_div and press Enter.) | |
| > my_div [1] 3.478505 3.181981 .146460 | |
| Perseverance, that's the answer. | |
| 100% | = ~ |

| Console ~/ 🔊] USE UNITINK(TESTOIT , recursive = IKUE). | -0 |
|--|----|
| ose untital cesturi , recursive = troey. | ^ |
| > unlink("testdir", recursive = TRUE) | |
| You got it! | |
| | |
| Take nothing but results. Leave nothing but assumptions. That sounds like 'Take nothing but pictures. Leave nothing but footprints.' But it makes no sense! Surely ur eaders can come up with a better motto | |
| | |
| | |
| In this lesson, you learned how to examine your R workspace and work with the file system of your machine from within R. Thanks for playing! | |
| | |
| 100% | |
| Would you like to receive chadit for completing this course on Coursera.org? | |
| | |
| 1: Yes | |
| 2: No | |
| Selection: | _ |







```
Console ~/ ⋈
                                                                                                                           -0
   |-----
  94%
| Let's see if that got the job done. Print the contents of my_data.
                                                                 esendit
> my_data
  patient age weight bp rating test
      Bill
                       5 9
                                  13
                                         17
      Gina
                       6 10
                                  14
                                         18
3
              3
                       7 11
                                        19
     Kellv
                                  15
4
              4
                       8 12
                                         20
      Sean
                                  16
| You are really on a roll!
  _____
  97%
| In this lesson, you learned the basics of youking with two very important and common data | structures -- matrices and data frames. There is much more to learn and we'll be covering more | advanced topics, particularly with respect to data frames, in future lessons.
 100%
| Would you like to receive redit for completing this course on Coursera.org?
1: Yes
2: No
Selection: 2
```

```
Console ~/ 🙈
| You are quite good my friend!
                                       oResendir
 _____
| Which of the following evaluates to TRUE?
1: all(c(TRUE, FALSE, TRUE))
2: any(ints == 2.5)
3: all(ints == 10)
4: any(ints == 10)
Selection: 4
| Your dedication is inspiring!
 ogic in R. If you really want to see what you can do with logi
| That's all for this introducti
c, check out the control
| flow lesson!
| Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
Selection:
```

| Console ~/ ⇔ | -0 |
|---|----------|
| Sourcing your script | ^ |
| | |
| All that hard work is paying off! | |
| • 1 | |
| 96% | |
| | |
| You made your own binary operator! Let's test it out. Paste together the strings: 'I', 'love' | ١, |
| 'R!' using your new binary operator. | |
| > 'I' %p% 'love' %p% 'R!' | |
| [1] "I love R!" | |
| Excellent work! | |
| 2 CO | |
| | |
| ====================================== | |
| We've come to the end of our lescon no out there and write some great functions! | |
| | |
| | |
| | |
| ======================================= | |
| Would you like to receive credit for completing this course on Coursera.org? | |
| 1. 11- | |
| 1: No 2: Yes | |
| | |
| Selection: | ~ |
| | |

| Console ~/ ⇔ | |
|--|---|
| Excellent job! | ^ |
| | _ |
| ======================================= | |
| The only difference between previous examples and this one is that we are defining and using our on function right in the call to lapply(). Our function has no name and disappears as poor as lapply() is done using it. So | |
| called 'anonymous functions' can be very useful when one of R's built-in functions isn't an option. | |
| Trancetons can be very ascrar when one of it's barre in threetons is it can operon. | |
| | |
| | _ |
| ======================================= | |
| In this lesson, you learned how to use the powerful lapply() and sapply() functions to apply an operation over the elements of a list. In the next lesson, we'll take a look at some close relatives of lapply() and sapply(). | |
| | |
| 10 % | = |
| Would you like to receive credit for completing this course on Coursera.org? | |
| 1: No 2: Yes | |
| Selection: | V |

| Console ~/ ⇔ | |
|--|---|
| 1: 119.0 2: 1010.0 3: 5.00 4: 157.00 5: 56.00 | ^ |
| Selection: 5 | |
| Keep working like that and you'll get there! | |
| | = |
| In this lesson, you learned how to use vapply() is safer alternative to sapply(), which is most alpful when writing your own functions. You also learned how to use apply() to split your data into groups based on the value of some variable, then apply a function to each group. These runctions will come in handy on your quest to become a letter data analyst. | 1 |
| ··· | |
| 100% Would you like to receive credit for completing this course on Coursera.org? 1: Yes | = |
| 2: No | |
| Selection: | v |
| | |
| Console ~/ 	⇔ review of its contents. | ^ |
| | |
| = 92% | = |
| str() is actually a very general function that you can use on most of jects in R. Any time you want to understand the structure of something (a dataset, function, etc.), str() is a good place to start. | |
| | |
| | = |
| In this lesson, you learned how to get a feel or the structure and contents of a new dataset using a collection of simple and useful functions. Taking the time to to this upfront can save you time and frustration later of in your analysis. | |
| | |
| 100% | = |
| Would you like to rice; e credit for completing this course on Coursera.org? | |
| | |
| 1: Yes 2: No | |

| Console ~/ ⇔ | -6 | 7 |
|---|-------|---|
| m at work, but that's a | | ^ |
| lesson for another day! | | |
| | | |
| ••• | | |
| | | |
| ======== 94% | | |
| 1 34% | | |
| All of the standard probability distributions are built into R, i cluding exponential (rexp()), | , chi | |
| -squared (rchisq()), | | |
| gamma (rgamma()), Well, you see the pattern. | | |
| | | |
| | | |
| 0.8 | | |
| | | |
| | | |
| Simulation is practically a field of its on and we've only skimmed the surface of what's poss- | ible. | |
| I encourage you to | | |
| explore these and other functions further on your own. | | |
| | | |
| | | |
| | | |
| ======= 100% | | |
| | | |
| Would you like to receiveredit for completing this course on Coursera.org? | | |
| 1: No | | |
| 2: Yes | | |
| 2. 10 | | |
| Selection: | | |
| · · · · · · · · · · · · · · · · · · · | | ٧ |

| Console ~/ 💫 | -6 |
|--|----|
| | ^ |
| | |
| ======== 94% | |
| Use difftime(Sys.time(), t1, units = 'days') to find the amount of time in DAYS that has passed s ce you created t1. | in |
| <pre>> difftime(Sys.time(), t1, units = 'days') Time difference of 0.005582975 days</pre> | |
| Great job! | |
| 97% | = |
| In this lesson, you learned how to work with dates and times in R. While it is important to undersand the basics, if you find yourself working with dates and times often, you may want to check out the lubridate package y Hadley Wickham. | |
| | |
| 100% | == |
| Would you like to receive credit for completing this course on Coursera.org? | |
| 1: Yes 2: No | |
| Selection: | V |
| Console ~/GitHub/Programacion_Actuarial_3/ | _@ |
| is best used by just passing in a single vector. | ^ |
| | |
| | |
| | |
| 96% | |
| Use hist() with the vector mtcars\$mpg to create a histogram. | |
| <pre>> hist(mtcars\$mpg)</pre> | |
| That's a job well done! | |
| | |
| ======================================= | |
| In this lesson, you learned how to work with base graphics in R. The best place to go from here is to study the ggplot2 | 5 |
| package. If you want to explore other ele encs of base graphics, then this web page (http://www.ling.upenn.edu/~joseff/rctucy/week4.html) provides a useful overview. | |
| | |
| | |
| | == |
| 100% | == |
| 100% Would you like to receive credit for completing this course on Coursera.org? | == |
| | == |