> library(swirl)

| Hi! Type swirl() when you are ready to begin.

> swirl()

| Welcome to swirl! Please sign in. If you've been here before, use the same name as you did then. If

| you are new, call yourself something unique.

What shall I call you? Stephen

| Would you like to continue with one of these lessons?

1: R Programming Workspace and Files

2: No. Let me start something new.

Selection: 1

| List all the files in your working directory using list.files() or dir().

> list.files()

[1] "04 - Exploratory Data Analysis.Rproj" "Course Project 1"

[3] "R Exercises with Swirl" "README.md"

[5] "Slides"

| Excellent job!

|================================ | 26%

| As we go through this lesson, you should be examining the help page for each new function. Check

| out the help page for list.files with the command ?list.files.

> ?list.files

| All that hard work is paying off!

|=================================== | 28%

| One of the most helpful parts of any R help file is the See Also section. Read that section for

| list.files. Some of these functions may be used in later portions of this lesson.

...

|====================================== | 31%

| Using the args() function on a function name is also a handy way to see what arguments a function

| can take.

...

|========================================= | 33%

| Use the args() function to determine the arguments to list.files().

> args(list.files)

function (path = ".", pattern = NULL, all.files = FALSE, full.names = FALSE,

recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE,

no.. = FALSE)

NULL

| Your dedication is inspiring!

|============================================ | 36%

| Assign the value of the current working directory to a variable called "old.dir".

> old.dir <- getwd()

| That's a job well done!

|=============================================== | 38%

| We will use old.dir at the end of this lesson to move back to the place that we started. A lot of

| query functions like getwd() have the useful property that they return the answer to the question

| as a result of the function.

...

|================================================== | 41%

| Use dir.create() to create a directory in the current working directory called "testdir".

> dir.create(testdir)

Error in dir.create(testdir) : object 'testdir' not found

> dir.create("testdir")

| You are quite good my friend!

|====================================================== | 44%

| We will do all our work in this new directory and then delete it after we are done. This is the R

| analog to "Take only pictures, leave only footprints."

...

|========================================================= | 46%

| Set your working directory to "testdir" with the setwd() command.

> setwd(testdir)

Error in setwd(testdir) : object 'testdir' not found

> setwd("testdir")

| You are doing so well!

|============================================================ | 49%

| In general, you will want your working directory to be someplace sensible, perhaps created for the

| specific project that you are working on. In fact, organizing your work in R packages using RStudio

| is an excellent option. Check out RStudio at http://www.rstudio.com/

...

|=============================================================== | 51%

| Create a file in your working directory called "mytest.R" using the file.create() function.

> file.create("mytest.R")

[1] TRUE

| Great job!

|================================================================== | 54%

| This should be the only file in this newly created directory. Let's check this by listing all the

| files in the current directory.

> list.files()

[1] "mytest.R"

| Excellent job!

|===================================================================== | 56%

| Check to see if "mytest.R" exists in the working directory using the file.exists() function.

> file.exists("mytest.R")

[1] TRUE

| You are really on a roll!

|========================================================================= | 59%

| These sorts of functions are excessive for interactive use. But, if you are running a program that

| loops through a series of files and does some processing on each one, you will want to check to see

| that each exists before you try to process it.

...

|============================================================================ | 62%

| Access information about the file "mytest.R" by using file.info().

> file.info("mytest.R")

size isdir mode mtime ctime atime exe

mytest.R 0 FALSE 666 2017-10-17 13:59:02 2017-10-17 13:59:02 2017-10-17 13:59:02 no

| You got it!

|=============================================================================== | 64%

| You can use the $ operator --- e.g., file.info("mytest.R")$mode --- to grab specific items.

...

|================================================================================== | 67%

| Change the name of the file "mytest.R" to "mytest2.R" by using file.rename().

> file.rename("mytest.R", "mytest2.R")

[1] TRUE

| You got it!

|===================================================================================== | 69%

| Your operating system will provide simpler tools for these sorts of tasks, but having the ability

| to manipulate files programatically is useful. You might now try to delete mytest.R using

| file.remove('mytest.R'), but that won't work since mytest.R no longer exists. You have already

| renamed it.

...

|======================================================================================== | 72%

| Make a copy of "mytest2.R" called "mytest3.R" using file.copy().

> file.copy("mytest2.R", "mytest3.R")

[1] TRUE

| Excellent work!

|=========================================================================================== | 74%

| You now have two files in the current directory. That may not seem very interesting. But what if

| you were working with dozens, or millions, of individual files? In that case, being able to

| programatically act on many files would be absolutely necessary. Don't forget that you can,

| temporarily, leave the lesson by typing play() and then return by typing nxt().

...

|=============================================================================================== | 77%

| Provide the relative path to the file "mytest3.R" by using file.path().

> file.path("mytest3.R")

[1] "mytest3.R"

| You got it!

|================================================================================================== | 79%

| You can use file.path to construct file and directory paths that are independent of the operating

| system your R code is running on. Pass 'folder1' and 'folder2' as arguments to file.path to make a

| platform-independent pathname.

> file.path("folder1", "folder2")

[1] "folder1/folder2"

| You got it right!

|===================================================================================================== | 82%

| Take a look at the documentation for dir.create by entering ?dir.create . Notice the 'recursive'

| argument. In order to create nested directories, 'recursive' must be set to TRUE.

> ?dir.create

| Nice work!

|======================================================================================================== | 85%

| Create a directory in the current working directory called "testdir2" and a subdirectory for it

| called "testdir3", all in one command by using dir.create() and file.path().

> dir.create(file.path("testdir2", "testdir3"), recursive = TRUE)

| You nailed it! Good job!

|=========================================================================================================== | 87%

| Go back to your original working directory using setwd(). (Recall that we created the variable

| old.dir with the full path for the orginal working directory at the start of these questions.)

> setwd(old.dir)

| You nailed it! Good job!

|============================================================================================================== | 90%

| It is often helpful to save the settings that you had before you began an analysis and then go back

| to them at the end. This trick is often used within functions; you save, say, the par() settings

| that you started with, mess around a bunch, and then set them back to the original values at the

| end. This isn't the same as what we have done here, but it seems similar enough to mention.

...

|================================================================================================================== | 92%

| After you finish this lesson delete the 'testdir' directory that you just left (and everything in

| it)

...

|===================================================================================================================== | 95%

| 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 our readers can come up with

| a better motto . . .

...

|======================================================================================================================== | 97%

| 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 credit for completing this course on Coursera.org?

1: Yes

2: No