What shall I call you? Steven

| Please choose a course, or type 0 to exit swirl.

1: R Programming

2: Take me to the swirl course repository!

Selection: 1

| Please choose a lesson, or type 0 to return to course menu.

1: Basic Building Blocks 2: Workspace and Files 3: Sequences of Numbers

4: Vectors 5: Missing Values 6: Subsetting Vectors

7: Matrices and Data Frames 8: Logic 9: Functions

10: lapply and sapply 11: vapply and tapply 12: Looking at Data

13: Simulation 14: Dates and Times 15: Base Graphics

Selection: 15

| | 0%

| One of the greatest strengths of R, relative to other programming languages, is the ease with which

| we can create publication-quality graphics. In this lesson, you'll learn about base graphics in R.

...

|== | 2%

| We do not cover the more advanced portions of graphics in R in this lesson. These include lattice,

| ggplot2 and ggvis.

...

|==== | 4%

| There is a school of thought that this approach is backwards, that we should teach ggplot2 first. See

| http://varianceexplained.org/r/teach\_ggplot2\_to\_beginners/ for an outline of this view.

...

|====== | 7%

| Load the included data frame cars with data(cars).

> data(cars)

| Nice work!

|======== | 9%

| To fix ideas, we will work with simple data frames. Our main goal is to introduce various plotting

| functions and their arguments. All the output would look more interesting with larger, more complex

| data sets.

...

|========== | 11%

| Pull up the help page for cars.

> ?cars

| You are doing so well!

|============ | 13%

| As you can see in the help page, the cars data set has only two variables: speed and stopping

| distance. Note that the data is from the 1920s.

...

|============== | 15%

| Run head() on the cars data.

> head(cars)

speed dist

1 4 2

2 4 10

3 7 4

4 7 22

5 8 16

6 9 10

| You nailed it! Good job!

|================ | 17%

| Before plotting, it is always a good idea to get a sense of the data. Key R commands for doing so

| include, dim(), names(), head(), tail() and summary().

...

|================== | 20%

| Run the plot() command on the cars data frame.

> plot(cars)

| You are really on a roll!

|==================== | 22%

| As always, R tries very hard to give you something sensible given the information that you have

| provided to it. First, R notes that the data frame you have given it has just two columns, so it

| assumes that you want to plot one column versus the other.

...

|====================== | 24%

| Second, since we do not provide labels for either axis, R uses the names of the columns. Third, it

| creates axis tick marks at nice round numbers and labels them accordingly. Fourth, it uses the other

| defaults supplied in plot().

...

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

| We will now spend some time exploring plot, but many of the topics covered here will apply to most

| other R graphics functions. Note that 'plot' is short for scatterplot.

...

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

| Look up the help page for plot().

> ?plot

| Excellent job!

|============================= | 30%

| The help page for plot() highlights the different arguments that the function can take. The two most

| important are x and y, the variables that will be plotted. For the next set of questions, include the

| argument names in your answers. That is, do not type plot(cars$speed, cars$dist), although that will

| work. Instead, use plot(x = cars$speed, y = cars$dist).

...

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

| Use plot() command to show speed on the x-axis and dist on the y-axis from the cars data frame. Use

| the form of the plot command in which vectors are explicitly passed in as arguments for x and y.

> plot(x = cars$speed, y = cars$dist)

| All that practice is paying off!

|================================= | 35%

| Note that this produces a slightly different answer than plot(cars). In this case, R is not sure what

| you want to use as the labels on the axes, so it just uses the arguments which you pass in, data

| frame name and dollar signs included.

...

|=================================== | 37%

| Note that there are other ways to call the plot command, i.e., using the "formula" interface. For

| example, we get a similar plot to the above with plot(dist ~ speed, cars). However, we will wait till

| later in the lesson before using the formula interface.

...

|===================================== | 39%

| Use plot() command to show dist on the x-axis and speed on the y-axis from the cars data frame. This

| is the opposite of what we did above.

> plot(x = cars$dist, y = cars$speed)

| You are really on a roll!

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

| It probably makes more sense for speed to go on the x-axis since stopping distance is a function of

| speed more than the other way around. So, for the rest of the questions in this portion of the

| lesson, always assign the arguments accordingly.

...

|========================================= | 43%

| In fact, you can assume that the answers to the next few questions are all of the form plot(x =

| cars$speed, y = cars$dist, ...) but with various arguments used in place of the ...

...

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

| Recreate the plot with the label of the x-axis set to "Speed".

> plot(x = cars$dist, y = cars$speed, xlab="Speed")

| Nice try, but that's not exactly what I was hoping for. Try again. Or, type info() for more options.

| Type plot(x = cars$speed, y = cars$dist, xlab = "Speed") to create the plot.

> plot(x = cars$speed, y = cars$dist, xlab="Speed")

| That's the answer I was looking for.

|============================================= | 48%

| Recreate the plot with the label of the y-axis set to "Stopping Distance".

>

> plot(x = cars$speed, y = cars$dist, xlab="Speed", ylab="Stopping Distance")

| You're close...I can feel it! Try it again. Or, type info() for more options.

| Type plot(x = cars$speed, y = cars$dist, ylab = "Stopping Distance") to create the plot.

> plot(x = cars$speed, y = cars$dist, ylab="Stopping Distance")

| You got it!

|=============================================== | 50%

| Recreate the plot with "Speed" and "Stopping Distance" as axis labels.

> plot(x = cars$speed, y = cars$dist, xlab="Speed", ylab="Stopping Distance")

| That's a job well done!

|================================================= | 52%

| The reason that plots(cars) worked at the beginning of the lesson was that R was smart enough to know

| that the first element (i.e., the first column) in cars should be assigned to the x argument and the

| second element to the y argument. To save on typing, the next set of answers will all be of the form,

| plot(cars, ...) with various arguments added.

...

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

| For each question, we will only want one additional argument at a time. Of course, you can pass in

| more than one argument when doing a real project.

...

|===================================================== | 57%

| Plot cars with a main title of "My Plot". Note that the argument for the main title is "main" not

| "title".

> plot(cars, main="My Plot")

| You are amazing!

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

| Plot cars with a sub title of "My Plot Subtitle".

> ?plot

> plot(cars, main="My Plot", sub="My Plot Subtitle")

| Give it another try. Or, type info() for more options.

| Type plot(cars, sub = "My Plot Subtitle") to create the plot.

> plot(cars, sub="My Plot Subtitle")

| You nailed it! Good job!

|========================================================= | 61%

| The plot help page (?plot) only covers a small number of the many arguments that can be passed in to

| plot() and to other graphical functions. To begin to explore the many other options, look at ?par.

| Let's look at some of the more commonly used ones. Continue using plot(cars, ...) as the base answer

| to these questions.

...

|=========================================================== | 63%

| Plot cars so that the plotted points are colored red. (Use col = 2 to achieve this effect.)

> plot(cars, col=2)

| You got it right!

|============================================================= | 65%

| Plot cars while limiting the x-axis to 10 through 15. (Use xlim = c(10, 15) to achieve this effect.)

> plot(cars, xlim = c(10,15))

| Your dedication is inspiring!

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

| You can also change the shape of the symbols in the plot. The help page for points (?points) provides

| the details.

...

|================================================================= | 70%

| Plot cars using triangles. (Use pch = 2 to achieve this effect.)

> plot(cars, pch = 2)

| That's the answer I was looking for.

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

| Arguments like "col" and "pch" may not seem very intuitive. And that is because they aren't! So,

| many/most people use more modern packages, like ggplot2, for creating their graphics in R.

...

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

| It is, however, useful to have an introduction to base graphics because many of the idioms in lattice

| and ggplot2 are modeled on them.

...

|======================================================================== | 76%

| Let's now look at some other functions in base graphics that may be useful, starting with boxplots.

...

|========================================================================== | 78%

| Load the mtcars data frame.

> mtcars

mpg cyl disp hp drat wt qsec vs am gear carb

Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4

Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4

Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1

Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1

Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2

Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1

Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4

Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2

Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2

Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4

Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4

Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3

Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3

Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3

Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4

Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4

Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4

Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1

Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2

Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1

Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1

Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2

AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2

Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4

Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2

Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1

Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2

Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2

Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4

Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6

Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8

Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

| Not exactly. Give it another go. Or, type info() for more options.

| Type data(mtcars) to load the data.

> data(mtcars)

| All that hard work is paying off!

|============================================================================ | 80%

| Anytime that you load up a new data frame, you should explore it before using it. In the middle of a

| swirl lesson, just type play(). This temporarily suspends the lesson (without losing the work you

| have already done) and allows you to issue commands like dim(mtcars) and head(mtcars). Once you are

| done examining the data, just type nxt() and the lesson will pick up where it left off.

...

|============================================================================== | 83%

| Look up the help page for boxplot().

> ?boxplot

| You are really on a roll!

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

| Instead of adding data columns directly as input arguments, as we did with plot(), it is often handy

| to pass in the entire data frame. This is what the "data" argument in boxplot() allows.

...

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

| boxplot(), like many R functions, also takes a "formula" argument, generally an expression with a

| tilde ("~") which indicates the relationship between the input variables. This allows you to enter

| something like mpg ~ cyl to plot the relationship between cyl (number of cylinders) on the x-axis and

| mpg (miles per gallon) on the y-axis.

...

|==================================================================================== | 89%

| Use boxplot() with formula = mpg ~ cyl and data = mtcars to create a box plot.

> boxplot(mpg ~ cyl, data = mtcars)

| Excellent work!

|====================================================================================== | 91%

| The plot shows that mpg is much lower for cars with more cylinders. Note that we can use the same set

| of arguments that we explored with plot() above to add axis labels, titles and so on.

...

|======================================================================================== | 93%

| When looking at a single variable, histograms are a useful tool. hist() is the associated R function.

| Like plot(), hist() is best used by just passing in a single vector.

...

|========================================================================================== | 96%

| Use hist() with the vector mtcars$mpg to create a histogram.

> hist(mtcars$mpg)

| You are amazing!

|============================================================================================ | 98%

| 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 package. If you want to explore other elements of base graphics, then this web page

| (http://www.ling.upenn.edu/~joseff/rstudy/week4.html) provides a useful overview.

...

|==============================================================================================| 100%

| Would you like to receive credit for completing this course on Coursera.org?

1: No

2: Yes