Hi and thanks for coming to this introduction to R and RStudio. I’m so glad you're here! I’m Sharon Machlis, formerly executive editor at Computerworld and now director of data and analytics for Computerworld's parent company, IDG Communications.

I’ve been obsessed with R ever since I discovered it while working on an investigative project about skilled worker visas in 2013. I’m currently writing a book for Taylor Francis CRC Press’s R series, about R for journalism.

My main goal in the next few hours is to give you a sample of what R can do for you when you want to work with data. Obviously, you can’t “learn R” in 3 hours. But you *can* get a taste of its power and flexibility, and learn some basic skills. My other goal is to get you leaving here thinking “I’d like to learn more.”

This session will have three parts.

The first part will show you some R and RStudio software basics.

Part 2 is what I call the eye candy part. I want to show you a couple of cool visualizations you can do with very little code. R does have a learning curve, so I want to make the case that it's worth the investment. But unfortunately for this part, I won’t have enough time to *explain* the code. I’ll need to ask you to take it on faith that it will be useful to just run the code and see what happens.

The final and meatiest part of our session will be using R to explore some data sets. Here I *will* be explaining what the code is doing and how it works.

OK enough talk, let’s get cracking. Open up RStudio.

You’ll see 4 panes. If you don’t, click the little square by the closed panes to open it. Let me give you a very quick tour.

The bottom left pane is your interactive console. Type some code in there and you get an immediate response. The little greater-than sign is your prompt, that shows you where you can type.

Type some math in there – 52 + 112, or whatever you want – and you’ll get an answer back. That [1] next to the answer just means that it’s item number 1. You can sometimes have a LOT of items come back as an answer, so the start of each row tells you what number item it’s on.

The top left pane is your scripting panel. If you’re writing a lot of lines of code, you don’t want to have to keep typing it in line by line, without saving it. If you go to the menu and select File > New File > R Script, and then type your equation in there … save it as test.R, then click the Source button, R will run the entire script file.

There are 2 other ways to run code up there. One is line by line, so if you put your cursor anywhere on the line and hit control-enter, that line will execute. Another way is to define a portion of code in there, just like you’d define something before copying and pasting, and then hit control-enter.

Top right pane will show what objects and data are in your current session memory. There’s also a tab to show your command history. Click on the History tab, and you should see what you’ve typed in so far.

The bottom right pane has some other utilities. The Files tab is like a mini file explorer. There’s also a help tab for viewing help files, a Plots tab for your visualizations, and a couple of other things.

But enough with the grand tour, let’s start running some code.

In the bottom right pane, find the intro.R file, click on it, and it should open in the top left pane. And we’re going to run some code, line by line.

Put your cursor on the first 21 \* 3 line, hit control enter. You’ll get the answer or 63 and your cursor jumps to the next line.

We can save values in variables. Here I’m story 12 in x and 3 and y, and then multiplying the variables, which does the same thing as 21 times 3.

If you are wondering, what is that strange less than/minus sign arrow looking thing? It’s R’s way of saying “equals”. More accurately, it’s R’s “assignment operator,” saying the value of 21 is assigned to x. If you replaced that with an equals sign, that will work in most cases. But it’s the overwhelming custom in R to use the assignment operator, even when equals would work.

OK, let’s do some eye candy. Scroll down past all the green-colored lines that start with the pound sign – those are just comments, which you can put in to explain your code. They’re not actually R, and R won’t run them.

Control click on library(quantmod). That loads an external package called quantmod – one with extra functions not in base R – into your working session.

Next control click on that JPMorganChase getSymbols line and look at the top right pane. There’s an object called JPMorganChase in there. Click on it, and you’ll see what’s inside.

That is every daily stock price for JP Morgan since 2007.

But wait, there’s more. Run the next line.

[Go to intro.R script for more narrative]

[After intro.R, go to flight\_delays.R script]

🔑 filter – You only want rows that meet certain criteria. Campaign contributions over $100 to certain candidates. Bridges that haven’t been inspected for more than a year.

🔑 select – This picks the *columns* you want. A data set might have 40 or 50 columns, but for a certain analysis, you only need 6 of them.

🔑mutate – Add a new calculated column. Campaign expenditures divided by a candidate’s

🔑arrange

🔑summarize

🔑group\_by