SUPER DATASCIENCE
MAKING THE COMPLEX SIMPLE

SDS PODCAST EPISODE 790: OPEN-SOURCE LIBRARIES FOR DATA SCIENCE AT THE NEW YORK R CONFERENCE

Jon Krohn: 00:02 This is Five Minute Friday on open-source libraries for data science.

00:19 Welcome back to the Super Data Science Podcast. Today's episode was filmed live on stage in May at the New York R Conference. I hosted a panel that consisted of four data science icons, namely Drew Conway, who's now head of data science for private investments at Two Sigma, one of the world's largest hedge funds. Jared Lander, who's adjunct faculty on statistical programming at Columbia University. And author of the bestselling book are is for Everyone. Emily Zabor, who's a biostatistician at the Cleveland Clinic's department for Quantitative Health, Sciences, and JD Long, VP of Portfolio Solutions at Renaissance Reinsurance. Today's episode will primarily be of interest to hands-on practitioners. In today's episode, I ask my four panelists to answer a question for me about their favorite open-source software library for data science. Ready? Let's jump right into that panel.

01:09 Before I call anyone by name, I'll give you a chance to work on this a little bit. So what is a notable open-source library? Maybe an R, maybe not. This is also about the open statistical programming meetup, so we can broaden things a bit. But what is a notable open-source package that you feel like people should know about, either because you love it so much or you hate it so much or it's new and you can't stop thinking about it and we'll go in the same order that we have been going. We'll start with Drew.

Drew Conway: 01:39 Sure. Well, I guess in the spirit of this being a retrospective panel, I won't talk about a new package. But in the days when I was running R meetup, and particularly in my years as a graduate student, I spent a lot of time working in kind of graphs and thinking about social networks, and those days was when you could get a lot of data about social networks and it was the early days of it.

02:03 And I happened in those days as well to be a kind of contributor to a Python package called NetworkX, which if you know is still quite popular, if you do that kind of work, it's still great. But I wanted to be doing things in R, but I didn't really... There wasn't NetworkX in R, So I finally found this package called igraph, which had everything I wanted in it. And again, in the spirit of meeting and learning about the community, I had so many questions about how to do the things that I had been doing in NetworkX and I got connected to the authors of that, one of which was a fellow graduate student who happened to be at the University of Michigan, a guy named Mike Bomarito, who, him and his partner now have gone on and done a bunch of really interesting work in generative AI around the law.

02:45 They're both lawyers now, and they do LLMs in legal work. And so that package became a central part of what pulled me over to being a full-time R programmer for a while because it was the one thing that I couldn't do an R now I suddenly could do. And that was the thing, and it's still a very well-maintained package. So if anybody's doing that kind of work, I highly recommend it.

Jon Krohn: 03:05 igraph.

Drew Conway: 03:05 Yeah.

Jon Krohn: 03:05 Very cool.

Drew Conway: 03:06 Cool.

Jon Krohn: 03:08 Jared.

Jared Lander: 03:08 So there's this package called Segue, so I'll give a real answer, but... So Segue is actually a package written by JD 14 years ago, back when it was hard to do things automatically in AWS. The reason I want to say, not just because I want to make a good joke, when someone mentioned that package at the meetup, and JD was not there, people cheered for it. So it was awesome.

03:32 But I guess for me, I'm really, really into the Targets package. It's like Make but in R, which sort of lets you build a DAG out of your functions automatically, you have a bunch of functions that feed into each other, it creates a DAG and it only executes the part of the code that needs to be executed. So you have a part of the code that hasn't changed since the last time you ran it. It uses cached results. And this has saved so much time and just completely redone my workflow. It's been so incredibly helpful. So I think everyone should know Targets.

JD Long: 04:00 What is the equivalent of Targets in Python?

Jared Lander: 04:03 So after it came out of Targets, they came out with Snakemake.

JD Long: 04:07 Oh.

Jared Lander: 04:08 So Snakemake... so Targets is based on Make. It was actually originally Targets, there was a package called Drake, which was the R version of Make, but then they redid it and made it Targets and then someone in the Python community liked it. I think Dan Chen might've been involved and they made Snakemake.

JD Long: 04:21 So Snakemake?

Jared Lander: 04:23 Yeah, like a snake. Python is a snake and make for Make.

JD Long: 04:27 For a second there, I thought you said "Snakebake."

Jared Lander: 04:34 Oh, okay. So Snakemake, apparently that is the equivalent of Targets. I don't know if it has feature parity, which one does... Targets, you could run this on a cluster or remotely. I don't know what Snakemake does, but it's supposed to be very similar.

Jon Krohn: 04:43 I was hoping Snakemake would be a one-line command that would allow you play Snake in the command line.

Jared Lander: 04:48 Boo. That'll get get your run out here, I'm sure.

JD Long: 04:52 The Nokia version of Snake.

Jon Krohn: 04:54 Exactly, exactly. Emily?

Emily Zabor: 04:57 Oh yeah, I was trying to think about what's a non-glitzy package that I use all the time and the package that I know that I load every single time I open up an R project to start a new project for a collaborator is the Janitor package. I don't know if you guys are all using this, but I mean every time I read in a data set to do an analysis, I run my remove empty and my clean names at minimum to get those couple of things like exactly where I know I need them because my files with my data names, that are the first tumor volume measure that I ever took in NG per ml parentheses, I collected this by hand. It takes care of that for me.

Jon Krohn: 05:39 Nice. Cleaning things up with Janitor.

05:41 And JD?

JD Long: 05:43 The most boring answer ever, and it's dplyr.

Jon Krohn: 05:45 I knew it.

JD Long: 05:48 It's just such good syntax that just works.

Jon Krohn: 05:50 Hadley left already.

JD Long: 05:52 I'm kissing his [beep] for nothing. That's a shame.

Drew Conway: 05:55 We all wanted to say it.

JD Long: 05:57 I am picking up a low-lying fruit here. The thing I like about it is I work with business folks who don't code at all, just Excel. They can look at the dplyr code that I write and talk to me about it like, oh, I think we should maybe do what? They just look at that and it's intuitive. It's that they probably couldn't write it from scratch, but they can parse it when they look at it and they get some of the if-then syntax is a little tricky. But after I explained it, they're like, "Oh, okay, cool." So I mean I think that's a testament to a clearly written high-level language when non-users can look at it, and understand generally what's going on.

Jon Krohn: 06:34 All right, that's it for today's concise episode on favorite open-source libraries for data science if you liked it. We'll be back shortly aiming for episode number 794 for another question, more open-ended to all four panelists at the New York R Conference. So look out for that episode coming up. Again, that's episode number 794.

06:59 Until then, if you enjoyed today's episode of the show, consider supporting it by sharing it with someone who might like it, reviewing the episode on your favorite podcasting platform. Subscribing of course, if you're not already a subscriber, but most importantly, just keep on listening. Until next time, keep on rocking it out there. And I'm looking forward to enjoying another round of the Super Data Science Podcast with you very soon.