

Focus groups: physicist attitudes toward software

Jim Pivarski

Princeton University – DIANA

December 5, 2016

What

I conducted two directed conversations— focus groups— with young physicists to find out how they use software and what they like and dislike about it.

Why

Some comments made among software developers seemed, to me, to be wildly out of touch with physicists' concerns or reflected the concerns of physicists who are no longer actively analyzing data (fingers on the keyboard).

What

I conducted two directed conversations— focus groups— with young physicists to find out how they use software and what they like and dislike about it.

Why

Some comments made among software developers seemed, to me, to be wildly out of touch with physicists' concerns or reflected the concerns of physicists who are no longer actively analyzing data (fingers on the keyboard).

But then, maybe I'm the one who's out of touch.

Is there a way to determine this objectively?

Talking to a friend who is now a sociologist:

Me: I'm thinking of creating a survey.

Her: You don't know enough to do a survey yet. Start with a focus group.

What's a focus group?

- ▶ Closed-door, directed conversation, led by a neutral moderator whose primary goal is to keep a minority from dominating the group and to stay on topic.
- ▶ Transcribed: anonymous members' quotes are the data.
- ▶ Encourages exploration: goal is to learn something *qualitatively new*.
- ▶ To minimize group effect, do independent trials.

August 29: 5 physicists, August 31: 5 physicists (color code)

- ▶ Requirement to join: must be *actively* analyzing data.
- ▶ Happenstance: all late grad school or postdoc, CMS, at Fermilab's LPC. Self-selected from 15 invitations.
- ▶ 1 hour for 3 questions, only two covered (both sessions).
- ▶ I moderated, recorded, and later transcribed.

1. What software do you use for analysis now? [subquestions]
2. How much time do you spend on [long list, broken down]?
3. Would you learn a new language if [breakdown of pros/cons]?

1. What software do you use for analysis now? [subquestions]
 - ▶ *Extremely* varied, surprising to me and each other.
 - ▶ However, ROOT was predominant: “You can’t escape it. Unless you end up dumping all your data to JSON or something, and then you’re going to be sad.”
2. How much time do you spend on [long list, broken down]?
3. Would you learn a new language if [breakdown of pros/cons]?

1. What software do you use for analysis now? [subquestions]
 - ▶ *Extremely* varied, surprising to me and each other.
 - ▶ However, ROOT was predominant: “You can’t escape it. Unless you end up dumping all your data to JSON or something, and then you’re going to be sad.”
2. How much time do you spend on [long list, broken down]?
 - ▶ Unable to estimate or rank: “Everyone spends a lot of time on everything.”
3. Would you learn a new language if [breakdown of pros/cons]?

1. What software do you use for analysis now? [subquestions]
 - ▶ *Extremely* varied, surprising to me and each other.
 - ▶ However, ROOT was predominant: “You can’t escape it. Unless you end up dumping all your data to JSON or something, and then you’re going to be sad.”
2. How much time do you spend on [long list, broken down]?
 - ▶ Unable to estimate or rank: “Everyone spends a lot of time on everything.”
3. Would you learn a new language if [breakdown of pros/cons]?
 - ▶ 3 answered/2 answered on the Google Doc afterward.
 - ▶ Open to Julia’s 1-based indexing but not lack of classes.
 - ▶ Mixed on metaprogramming and immutable data.
 - ▶ Everyone drew the line at LISP syntax: “I’d quit my job if I had to do that all day.”

“It depends on what you’re doing. Recently, I’m using Excel and Pandas a lot because I’m monitoring the CMS run level 1 rate. So recently I’m using it quite a bit. It really depends on what the project is.”

“I use MATLAB fairly frequently for certain things.”

“It depends on what you’re doing. Recently, I’m using Excel and Pandas a lot because I’m monitoring the CMS run level 1 rate. So recently I’m using it quite a bit. It really depends on what the project is.”

“I use MATLAB fairly frequently for certain things.”

HiggsCombiner, ljet, Theta, GGNtuple

MATLAB, Mathematica, Excel, Pandas, Numpy, Bokeh, rootpy

Bash, Perl, Awk

Condor, Slurm, PBS, CRAB, LxBatch, GridControl (alternative to CRAB)

Igprof, Valgrind, gdb

NetBeans, Eclipse, XCode, Jenkins, Travis-CI

in addition to standard MC generators, CMSSW, and ROOT.

“Trying to get an IDE to cooperate with ROOT and CMSSW is just asking for pain and torture.”

“I know of people from UCSB that use Eclipse. I always wanted to invest the time to try to do the same thing. . . I tried four or five years ago— that was painful. It failed for me.”

“I used to use NetBeans. . . but now I’m using XCode to see all of my codes.”

“I have no simple ROOT/CMSSW example for continuous integration.”

“The ROOT developers are a lot better at being software developers than I am, but I feel like there’s a lot of bad design choices.”

“I’ve sort of forgotten by now what a learning curve it was to learn ROOT. It has so many gotchas and so many pieces of the design you might not anticipate or ever think to do.”

“I think also the consensus is that ROOT is very user-unfriendly.”

“While it can be hard to find something, ROOT’s implemented a *lot* of different data structures and functions.”

“One thing I would like to say that I like about ROOT, in addition to having very fast and flexible data access, is that the developers are fairly responsive to analysis needs, at least in the medium term. . . Of course, the downside is that the whole language is very poorly organized and the inheritance structure of most of the user interface classes doesn’t make sense.”

“Let’s be clear: ROOT 6 is backward-incompatible even though they said it wasn’t.”

“ROOT is the most leaky program in existence.”

“I’ll start by saying I’m very thankful to the big team of developers that has developed CMSSW... [praise]... That being said, I’d love to not have to look at it myself.

“I think something that might be consensus is that CMSSW or something like it is really required to funnel all of the knowledge from all the different parts of the collaboration into something runnable. But also that we try not to use it more than necessary.”

“We don’t use CMSSW and CRAB jobs; we use the FWLite version to just write C++.”

“Most people aren’t writing EDAnalyzers to do their final-step analysis.”

“I like to use PyROOT because the development time to write code... is very quick. Like, for me, it’s probably an order of magnitude faster than C++.”

“Something like Python is so much more attractive than something like C++.”

“C++ is a language that invites mistakes.”

“I’d say my C++ skills are somewhere in the collaboration— not the worst, not the best— but some of the code in CMSSW was written by people with way more appetite for C++ than I have. It can be hard to understand, just looking at the code, what the person was trying to achieve.”

“None of us are going to do a ten times speedup because we wrote better code.”

[I don't think this is true. I've found bigger factors than this in user's analysis code. Random twiddling might unearth factors of ten.]

“I've actually observed that generally whenever I publish a paper, the ntuples on which I base that paper are generally the *worst* version of the ntuples in terms of speed and size, compared to anything around there at either side in time, because it's getting worse and then I stop and rethink and slim it down, and then it gets worse again.”

“There’s another way of thinking about it. Currently, if you have a big farm of computers working for you, how long are you going to wait for an analysis iteration? I would say it’s like an hour. Maybe you can get through a handful of analysis iterations in a day. If it gets much slower than that, it’s like, sort of an intolerable setup.”

“An hour is an intolerable setup? Wow.”

“Or more. Yeah, if you need, like, two hours per iteration, that’s pretty tough. In a day’s work, you don’t make four tests.”

- ▶ My questions definitely wouldn't have worked as a survey, but the focus group raised issues I hadn't been thinking about, as desired.
- ▶ General takeaway:
 - ▶ Physicists are plastic: they'll adapt to any new thing if there's a point to it and it gets the job done.
 - ▶ They're taking it upon themselves to look for time-saving tools.
 - ▶ One group was strongly in favor of Python over C++. The other didn't bring it up.
 - ▶ They underestimate the possibility of code performance gains, though big factors would impact their lives.
- ▶ I'm planning another focus group in January to get feedback on the design of my Femtocode project.