Delphi's COVIDcast Project

Lessons Learned Building Statistical Software in Real Time

Alex Reinhart
Delphi Group, Carnegie Mellon University
August 8, 2022

What is statistical computing?

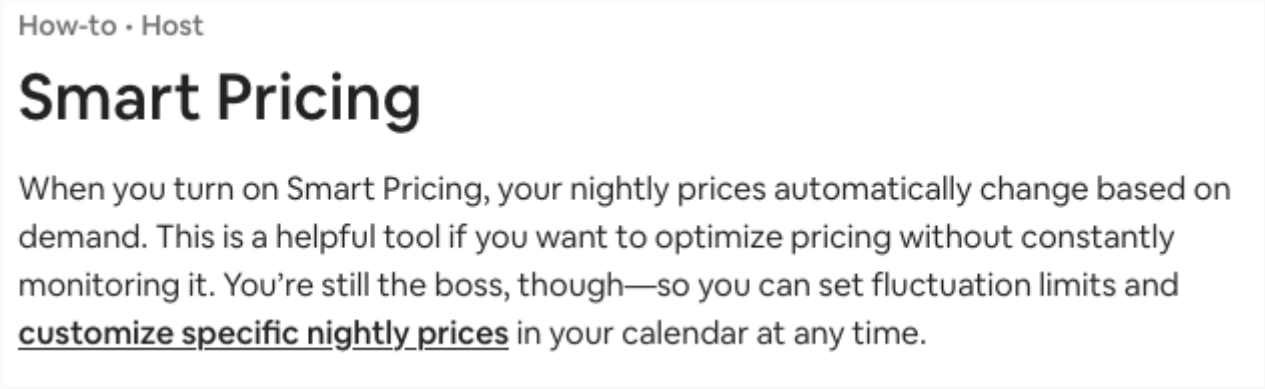
A few threads of development:

- Computational methods: Numerical optimization, MCMC, scalable algorithms
- Literate statistical programming: Reproducible reports, R Markdown, graphics & visualization
- Tools for programming: the Tidyverse, IDEs, new R packages

Or, in short, statistical computing is about statisticians using software.

But now lots of software uses statistics

- Tech companies use statistics & data science in products
- Businesses want models to make logistics and product decisions
- Data wrangling and fancy statistical methods get used in even the most mundane of products
- Statisticians in academia and industry are called upon to contribute to product development



(From Airbnb; see Bion et al, 2018)

How do we build statistical products?

- Writing software that uses statistics is different from writing statistical software
- The skills and methods required are distinct — and not widely appreciated
- Let's explore one statistical product to see those skills

Case study: COVIDcast

Delphi

- Since 2012, Delphi has developed "the theory and practice of epidemic forecasting, and its role in decision making"
- Led by Roni Rosenfeld and Ryan Tibshirani, with several participating faculty and graduate students
- Participated in annual CDC flu forecasting challenges, won several
- Named an Influenza Forecasting Center of Excellence by the CDC in 2019
- Published open code and data, including numerous influenza surveillance streams
- [I joined in April 2020]

https://delphi.cmu.edu/

Delphi's COVID-19 response

March 2020 saw a rapid expansion in Delphi and a change in goals

Now, with over 70 members, Delphi develops COVIDcast: data sources, maps, surveys, and code to support researchers, plus COVID forecasting

not everyone! →

Motivation

Imagine yourself in March 2020. How do you help public health officials make decisions when little data is available?

COVIDcast includes:

1. Code and infrastructure to obtain "indicators" daily—each indicator measures some signal relevant to COVID-19 in the United States
2. Unique relationships with healthcare and tech partners granting us access to indicators
3. A historical database of all indicators, including revision tracking, with >2.3b observations
4. An open API for requesting this data, with R and Python packages for easy access
5. An interactive visualization, built on the API, at delphi.cmu.edu/covidcast/
6. Forecasting and modeling work building on the data and API

This amounts to well over 200,000 lines of code (in R, Python, JavaScript, PHP, ...)

COVIDcast indicators

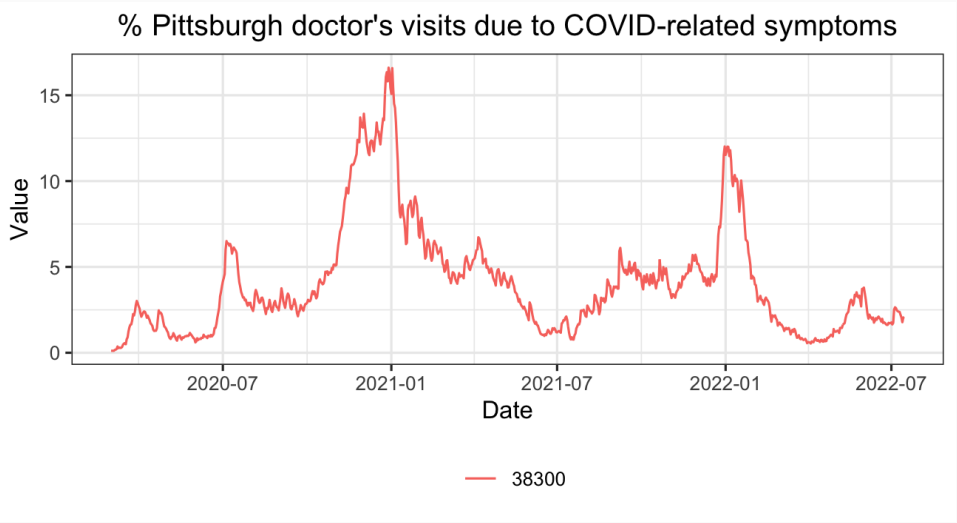
Freely available through the COVIDcast API, updated daily:

1. **Deaths** – from public reports
2. **Hospitalization** – from claims data and HHS data
3. **Case ascertainment** – from public reports, Quidel antigen tests, CTIS
4. **Outpatient visits** – from claims data
5. **Symptoms** – from CTIS, Google Search Trends
6. **General population** – SafeGraph mobility, plus CTIS

Most at the county level!

Full list: https://cmu-delphi.github.io/delphi-epidata/api/covidcast_signals.html

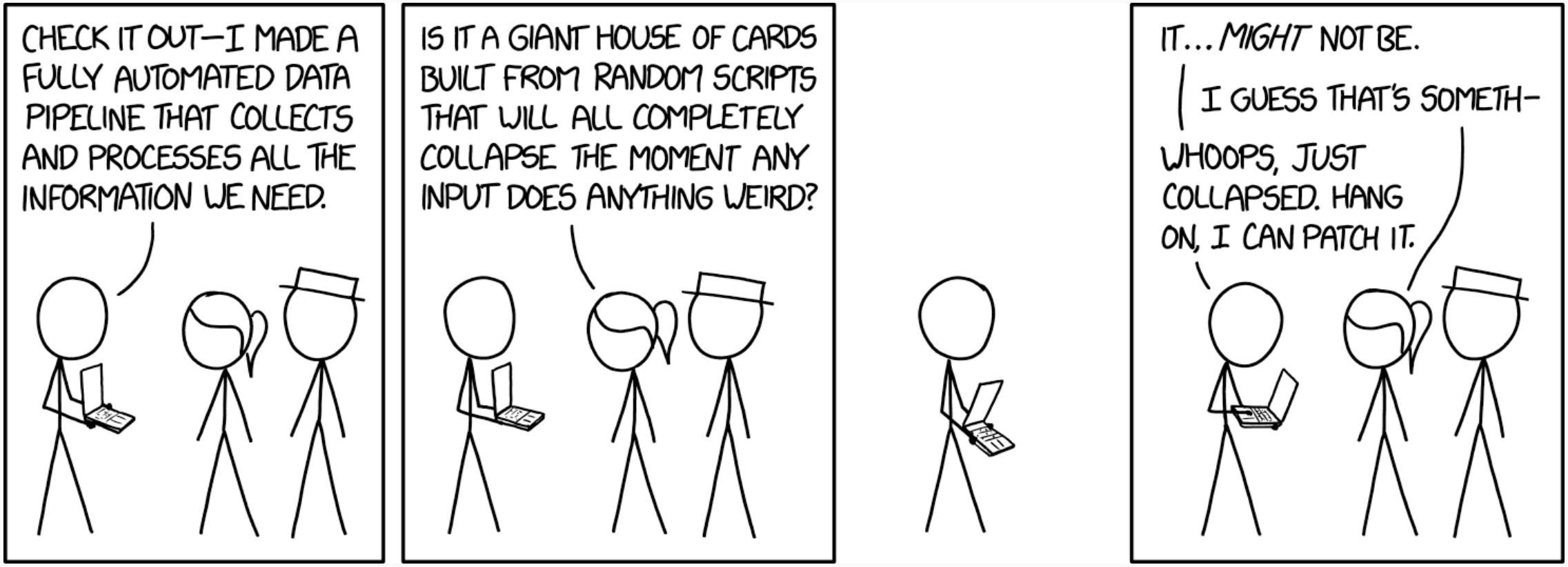
```
library(covidcast)
dv_pitt ← covidcast_signal(
  "doctor-visits", "smoothed_adj_cli",
  start_day = "2020-03-01", end_day = "2022-07-15",
  geo_type = "msa",
  geo_values = name_to_cbsa("Pittsburgh"))
plot(dv_pitt, plot_type = "line") +
  labs(title = "% Pittsburgh doctor's visits due to CO\
```



Statistical software engineering

Building a statistical product

- Making a statistical software product is more than just implementing statistical methods efficiently



xkcd #2054

Design requirements

To support forecasting, public health decision-making, and epidemic research, COVIDcast had to be:

1. Timely
2. Reliable
3. Accurate
4. Easy to change
5. Collaborative

This is statistical computing, but not as it's conventionally known

Timely

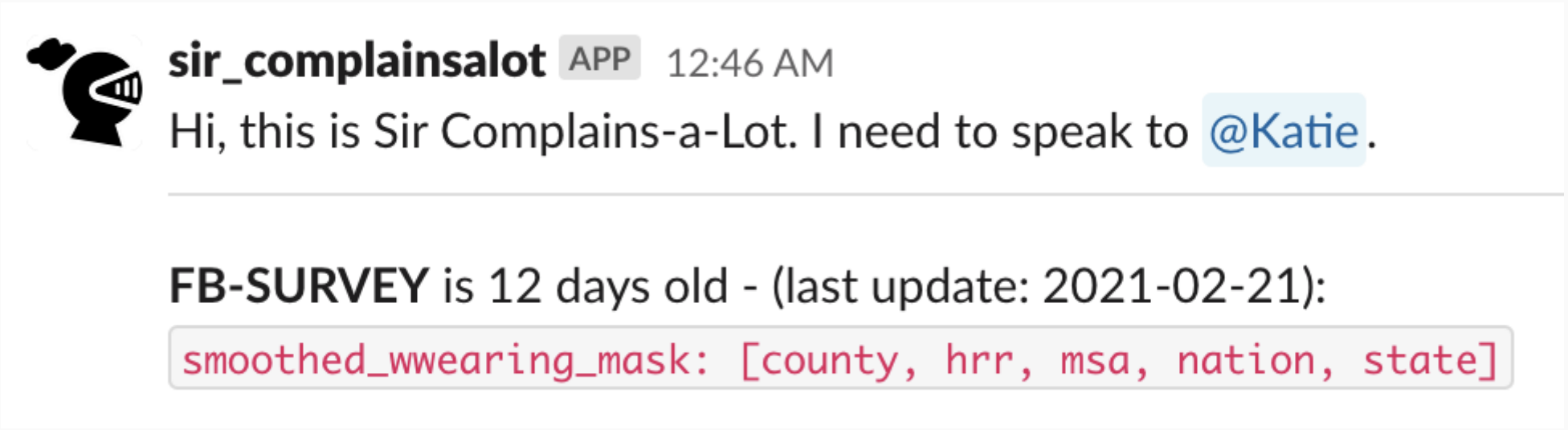
- New data must be posted daily, ideally within 1-3 days of events it records
- Aggregates and smoothing must be calculated for 3,000 counties, 400 metropolitan statistical areas, 450 Hospital Referral Regions, 50 states...
- Database must support efficient access to billions of rows

Timely

- COVID-19 Trends and Impact Survey data: Tens of thousands of responses daily that must be weighted and aggregated
- Naive implementation is roughly $O(n \times g \times d \times i)$, where n is number of responses, g is number of geographic regions, d is number of days to produce data for, i is number of aggregate indicators
- ...naive implementation had >24 hours of runtime per day
- "Folk wisdom" would have us vectorize, use Rcpp, and perhaps parallelize; but still $O(n \times g \times d \times i)$
- Profiling showed much of the runtime was in `filter()`
- But using binary search (`setkeyv` and `setindexv` from `data.table`) allows data filtering in log time: $O(n)$ to $O(\log n)$
- Profiling, data structures, and searching are bread-and-butter software engineering, but underused in statistical computing

Reliable

- A system that runs continuously without failure... and we should know when it fails
- Automation, error logging, and alerts are standard practice in industry
- Our initial pipeline automation: grad students with alarms
- After initial automation, it was still common for pipelines to silently fail to produce data, causing outages we didn't notice
- Hack solution: Sir Complains-a-Lot



Accurate

- Data we report should be accurate reflection of our source data
- Code review, unit testing, and continuous integration are becoming more popular in statistical computing
- Role in correctness is appreciated, but tests are important for long-term maintenance too

```
test_that("testing mix weights", {
  ## When all weights are identical and smaller than the minimum threshold, the
  ## minimum mixing mixes uniform with uniform and has no effect.
  weights ← rep(1, times = 200)

  mixed ← mix_weights(weights, s_mix_coef = 0.05, s_weight = 1L)
  expect_equal(mixed$weights,
               rep(1 / 200, 200))

  ## for intermediate version, check that mixing enforces the intended maximum
  for (k in seq(2, 5)) {
    mixed ← mix_weights(c(0.1, 0.1, 0.1, 0.1, 0.1, 0.5), s_weight = 1 / k,
                        s_mix_coef = 0.05)
    expect_lt(max(mixed$weights), 1 / k)
  }
}
```

Easy to change

- Code for data analysis reports is written and updated until the report is done
- Reproducibility ensures others can re-run the code later
- Statistical *products* are maintained and changed for months or years (the codebase of Theseus)
- Code must be written so it is easy to maintain and change later, perhaps by different people

Easy to change

- Maintaining thousands of lines of code over 2+ years means confronting the fallibility of your own memory:

```
# TODO: Determine if the fudge factors are really necessary
mix_coef ← if (max_weight ⩽ s_weight) {
  0
} else if (1/N > s_weight*0.999) {
  1
} else {
  (max_weight * N - 0.999 * N * s_weight + 1e-6) /
    (max_weight * N - 1 + 1e-6)
}
precoef ← mix_coef

# Enforce minimum and maximum.
if (mix_coef < s_mix_coef) { mix_coef ← s_mix_coef }
if (mix_coef > 1) { mix_coef ← 1 }
```

- Documentation and organization are more important than code cleverness
- Commit messages, GitHub Issues, narrative comments, unit tests, and READMEs are how you develop institutional memory

Easy to change

- Needs changed rapidly in early 2020 as we figured out what is useful
- Every data source started with its own complete codebase — small changes had to be coordinated across many places
- But now:

```
class GeoMapper:
    """Geo mapping tools commonly used in Delphi.

    The GeoMapper class provides utility functions for translating between different
    geocodes. Supported geocodes:
    - zip: zip5, a length 5 str of 0-9 with leading 0's
    - fips: state code and county code, a length 5 str of 0-9 with leading 0's
    - msa: metropolitan statistical area, a length 5 str of 0-9 with leading 0's
    - state_code: state code, a str of 0-9
    - state_id: state id, a str of A-Z
    - hrr: hospital referral region, an int 1-500
     ...
    """
```

- Thousands of lines of shared Python packages used by our codebase

Collaborative

- All this software needs to work together despite being written by dozens of grad students, faculty, research staff, contractors, and volunteers
- Code review is not just for correctness: it broadens knowledge
- Shared style and standards are important — if enforced
- But students get little collaborative software design experience

Preparing statistical software engineers

Software engineering

- Delphi was fortunate to ~~steal~~ hire skilled technical staff to help manage these processes
- ...and to borrow 13 engineers, designers, and managers from Google.org
- But statisticians in industry will face many of the same problems we did
- In industry, data scientists may work with software engineers
- But just as scientists should know a bit of statistics, we should know a bit of software engineering

Software engineering

- Computing courses need:
  - Tools like unit testing and version control
  - Basics of algorithms and data structures
  - Practice designing maintainable, well-structured software
  - Realistic long-term projects

JOURNAL OF STATISTICS AND DATA SCIENCE EDUCATION
2021, VOL. 29, NO. S1, S7–S15
https://doi.org/10.1080/10691898.2020.1845109

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Expanding the Scope of Statistical Computing: Training Statisticians to Be Software Engineers

Alex Reinhart and Christopher R. Genovese

Department of Statistics & Data Science, Carnegie Mellon University, Pittsburgh, PA

Wrapping up

Code is the product

- Statisticians and data scientists are increasingly developing products, not just reports or methods
- It's easy to say we should teach more things
- In the software engineering mindset, *code is the product*, not just a means to achieve other ends
- Courses can make code a deliverable along with reports/results
- But students also need real practice: capstone projects, internships
- Building statistical software products is not like a hackathon; it's like a marathon

Thank you

Thank you all for attending, and many thanks to

- the entire Delphi team (particularly Kathryn Mazaitis, ~~chief software wrangler~~ Engineering Lead)
- our colleagues at the University of Maryland and LMU Munich
- CMU Legal, Sponsored Programs, Communications, IT, and numerous staff
- Meta, Google, and Amazon Web Services
- Quidel
- Change Healthcare
- Qualtrics
- Centers for Disease Control and Prevention

Contact: https://delphi.cmu.edu, areinhar@stat.cmu.edu