Stitched

The new standard in Talent Analytics

Richard Washer

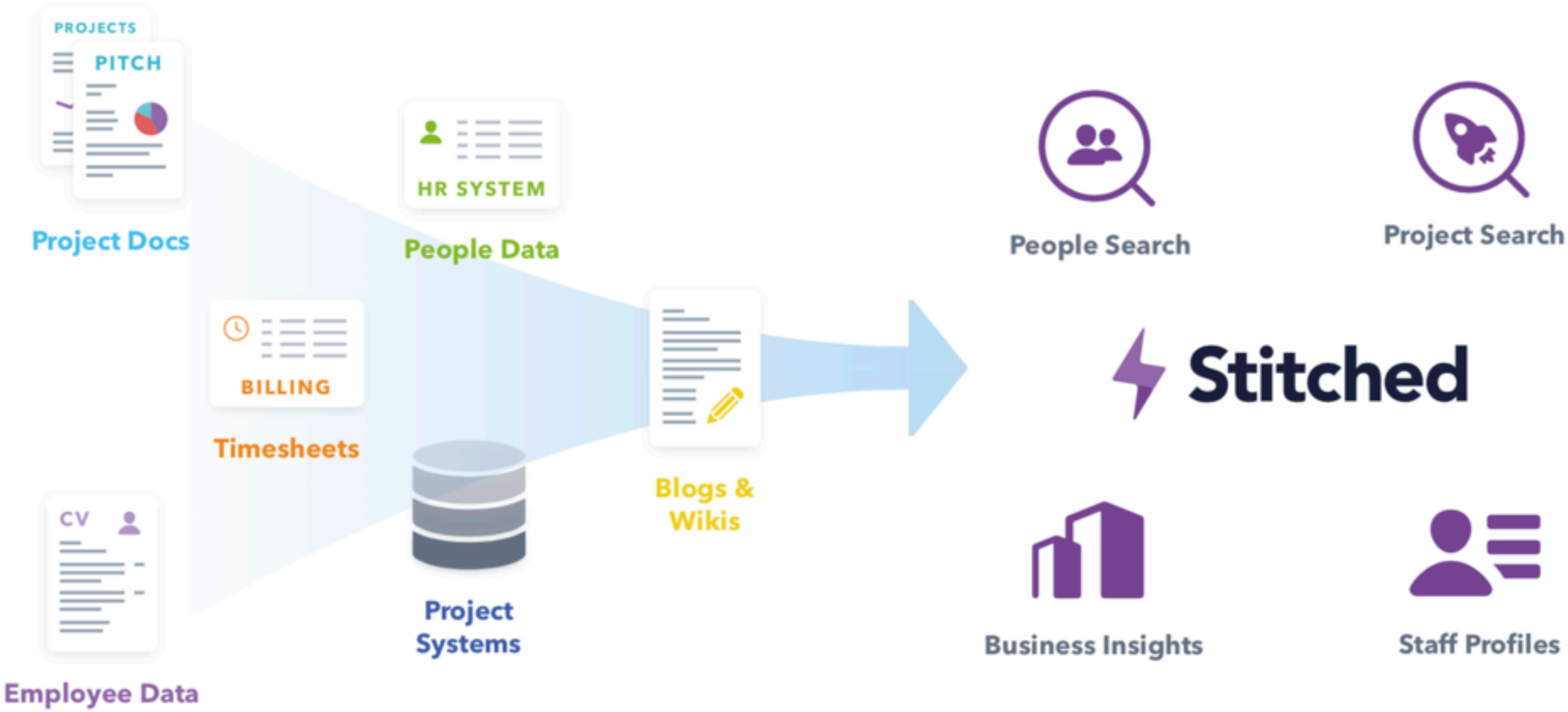
- rwasher.com
- github.com/rwasher
- twitter.com/richardwasher
- richard@stitched.io

- Alarmingly Kiwi (now London)
- ALL JS ALL THE TIME
- ~~new~~ Python()

WHY

- How to convince your company to let you use python for fun and profit
- Getting started on work with unclear strategy and product
- Making safe choices
- Various Python wins
- Always be shipping, if it isn’t in “production” then its vaporware

Stitched



PROJECTS

PITCH

Project Docs

HR SYSTEM

People Data

BILLING

Timesheets

CV

Employee Data

Project Systems

Blogs & Wikis

Stitched

People Search

Project Search

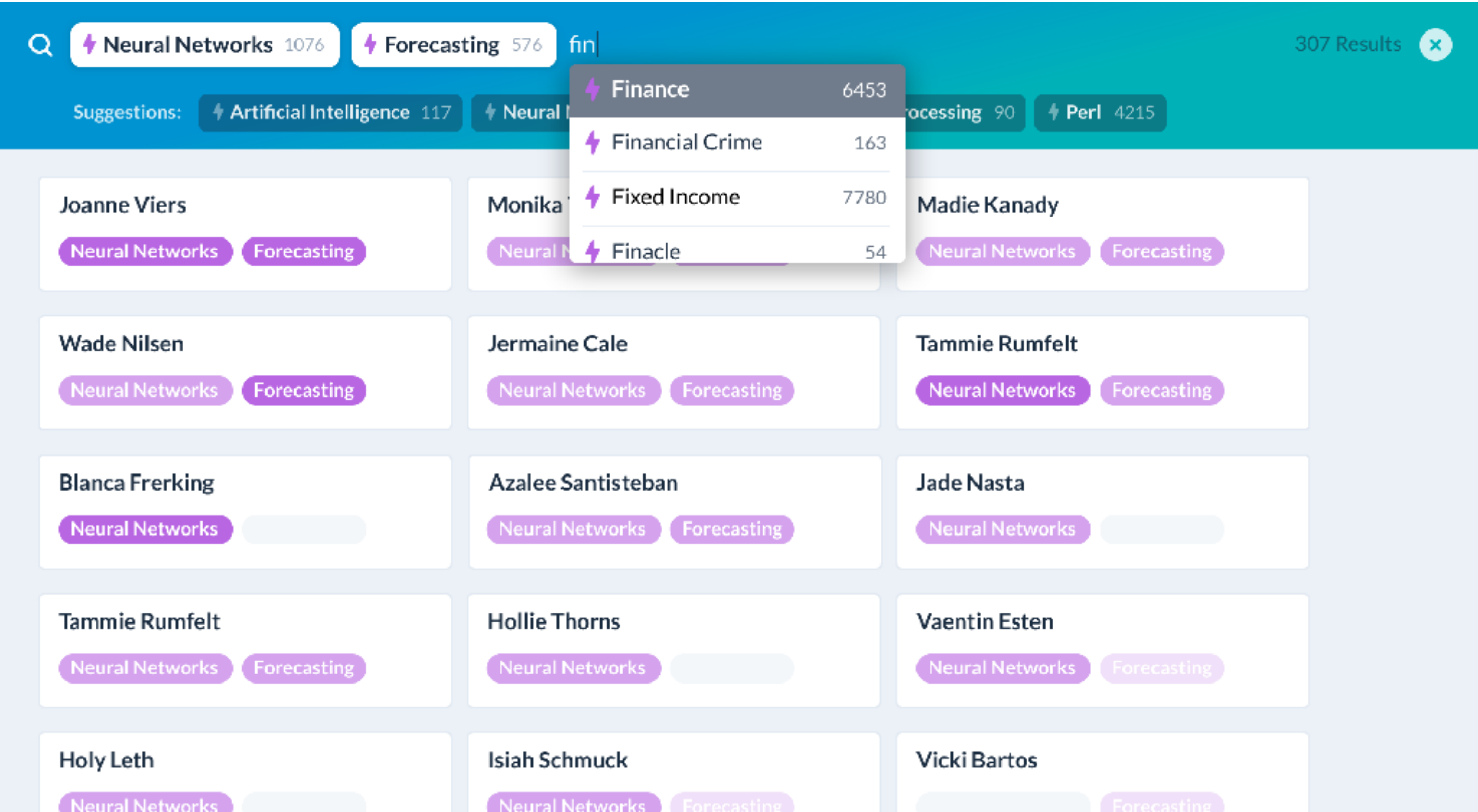
Business Insights

Staff Profiles

Stitched

- Natural language processing/machine learning
- Scan CVs, projects, timesheets - “Digital Footprint”
- Word2Vec, TF-IDF, tSNE, k-means
- “Word sense disambiguation for domain specific acronyms and homonyms”
- Search for people. Find people. Have a pint 🍺
- Have too much tech…

PRODUCT 1: SEARCH



Finance	6453
Financial Crime	163
Fixed Income	7780
Finacle	54

Joanne Viers — Neural Networks, Forecasting

Monika — Neural

Madie Kanady — Neural Networks, Forecasting

Wade Nilsen — Neural Networks, Forecasting

Jermaine Cale — Neural Networks, Forecasting

Tammie Rumfelt — Neural Networks, Forecasting

Blanca Frerking — Neural Networks

Azalee Santisteban — Neural Networks, Forecasting

Jade Nasta — Neural Networks

Tammie Rumfelt — Neural Networks, Forecasting

Hollie Thorns — Neural Networks

Vaentin Esten — Neural Networks, Forecasting

Holy Leth — Neural Networks

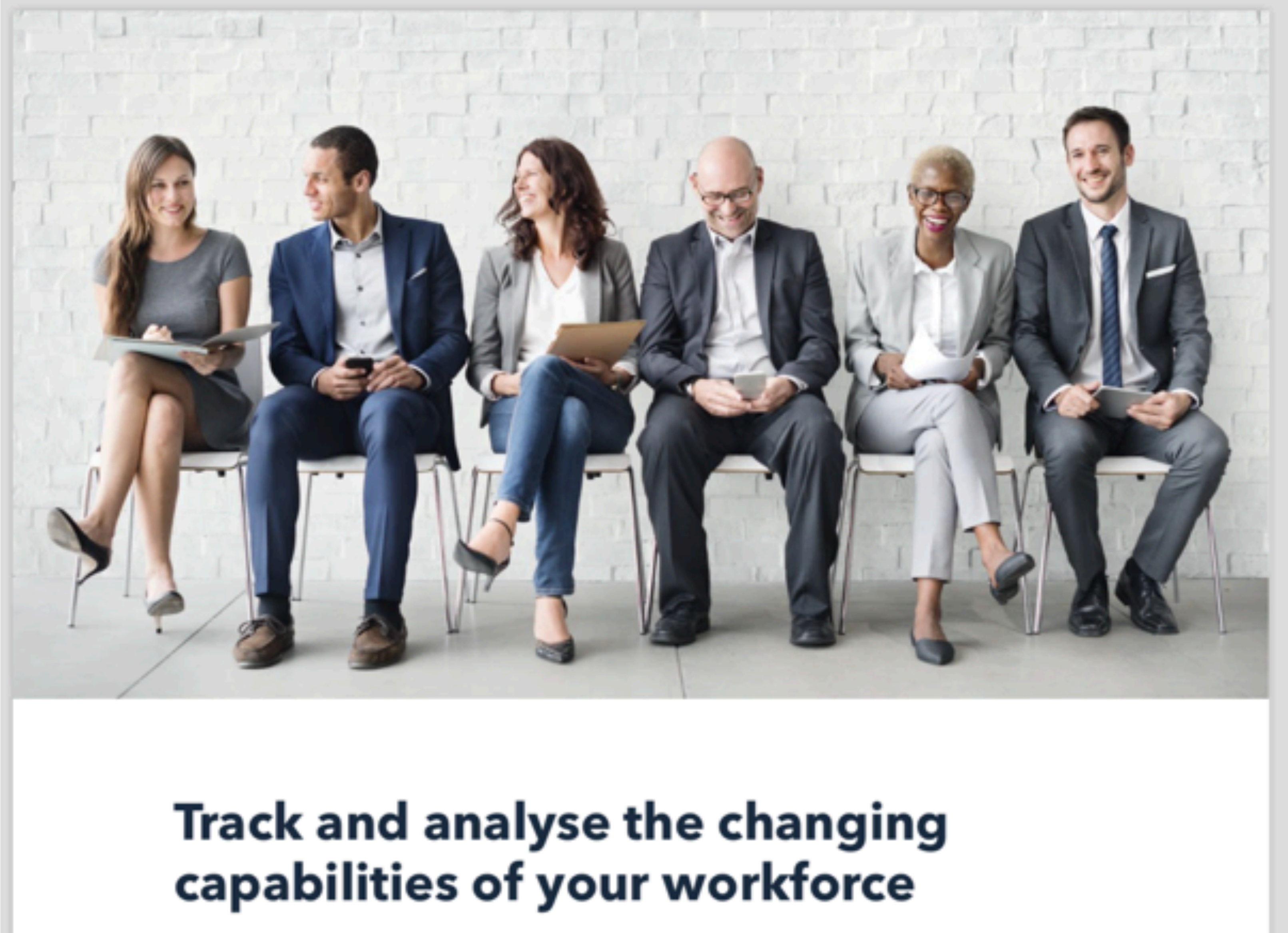
Isiah Schmuck — Neural Networks, Forecasting

Vicki Bartos — Forecasting

A NEW HOPE

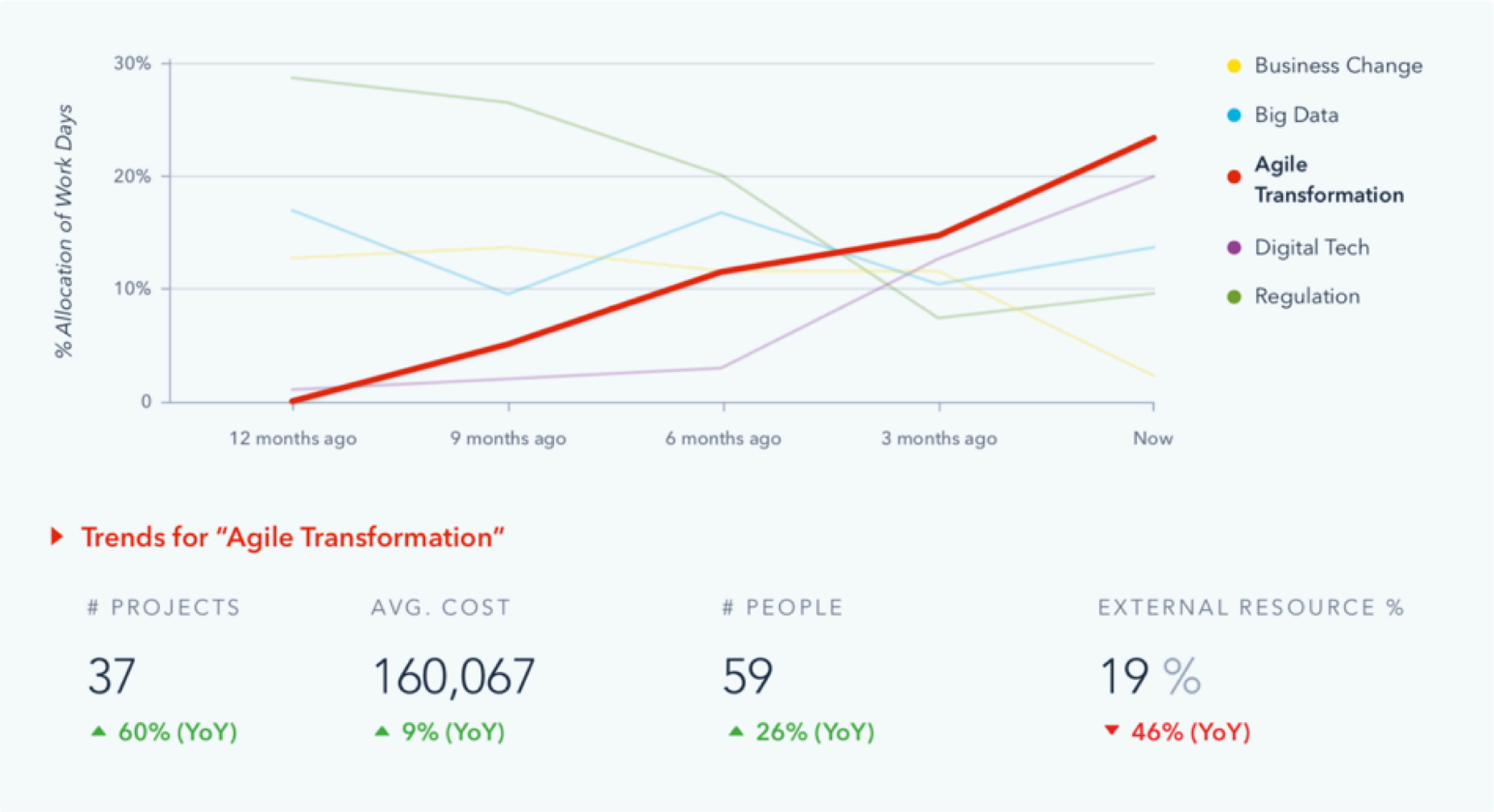
- Same data and underlying tech/API
- Targeted to “buyers”: C-suite, BAs, Upper Mgmt
- But **WHAT IS IT?**
- We’re not sure yet
- Something about visualising your organisation’s data
- How do we get started?
- **TELL**
- First 3-4 weeks, only resource is JS developer (Me)

A NEW HOPE



Track and analyse the changing capabilities of your workforce

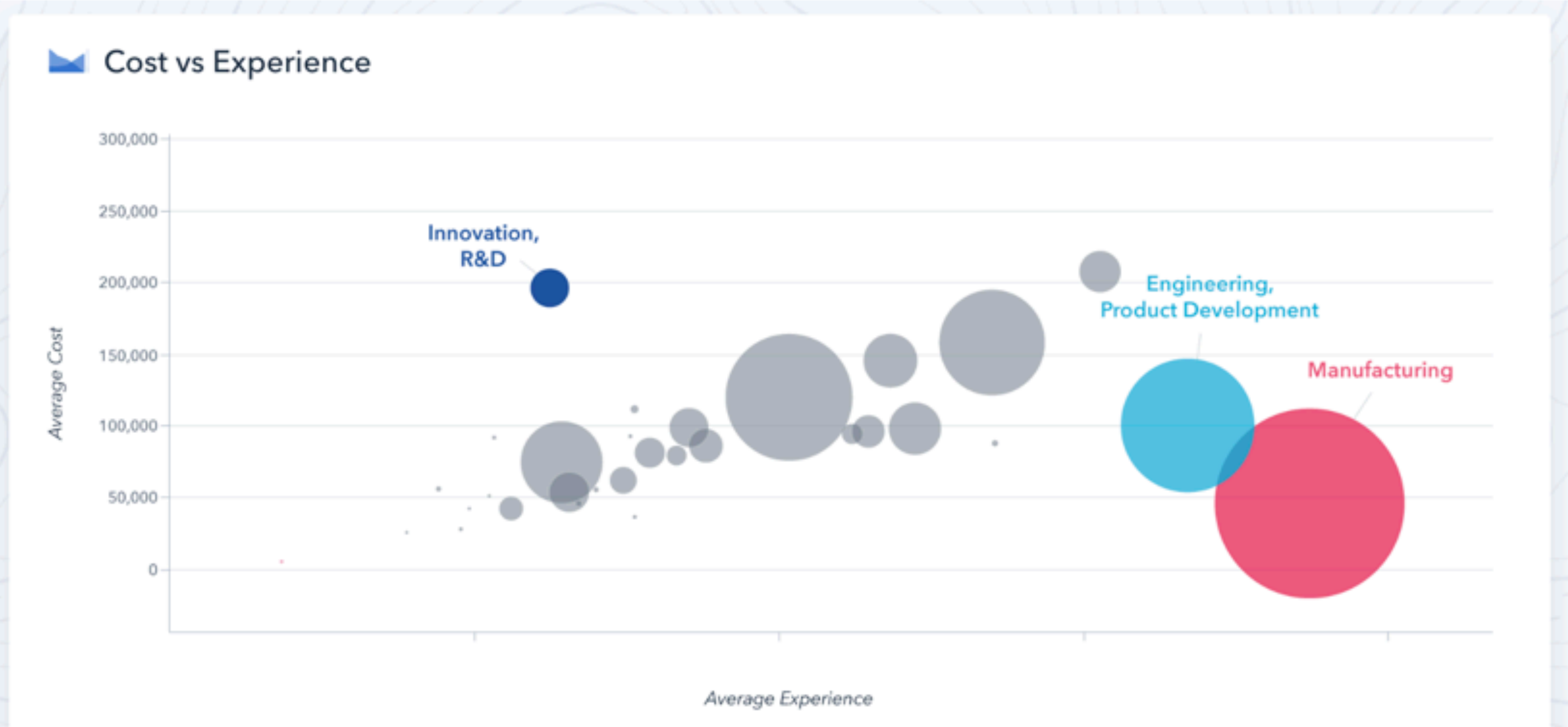
A NEW HOPE



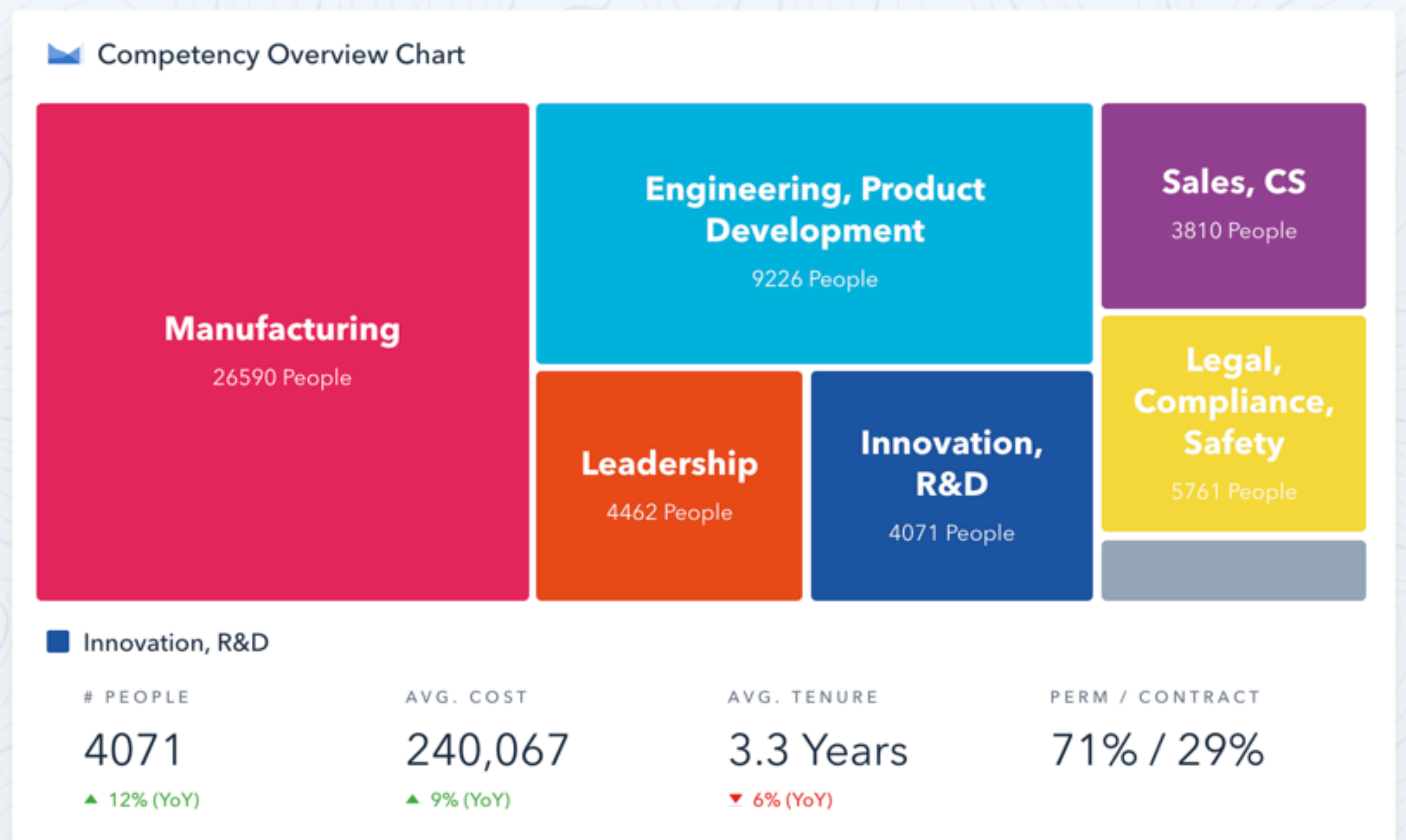
▸ Trends for "Agile Transformation"

# PROJECTS	AVG. COST	# PEOPLE	EXTERNAL RESOURCE %
37	160,067	59	19 %
▲ 60% (YoY)	▲ 9% (YoY)	▲ 26% (YoY)	▼ 46% (YoY)

A NEW HOPE


Cost vs Experience
300,000
250,000
200,000
150,000
100,000
50,000
0
Average Cost
Average Experience
Innovation, R&D
Engineering, Product Development
Manufacturing

A NEW HOPE



Competency Overview Chart

🟦 Innovation, R&D

# PEOPLE	AVG. COST	AVG. TENURE	PERM / CONTRACT
4071	240,067	3.3 Years	71% / 29%
▲ 12% (YoY)	▲ 9% (YoY)	▼ 6% (YoY)	

HOSPITAL PASS

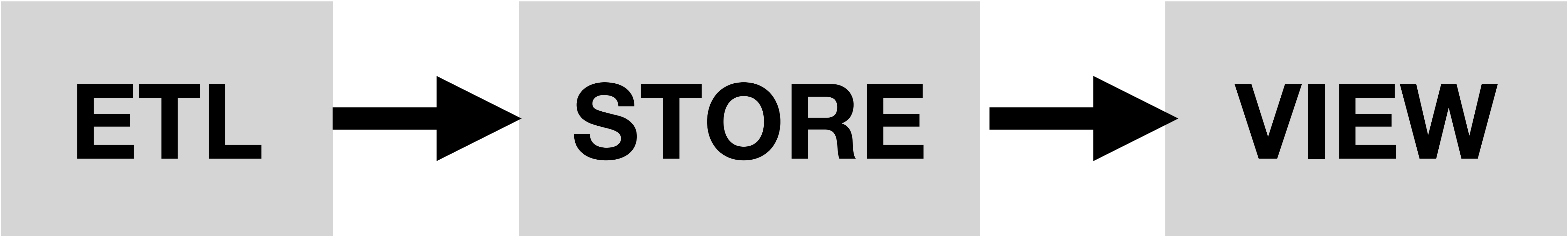
- No new endpoints (small tweaks maybe)
- Loosely defined value proposition
- No user stories
- How live is the data?
- Charts? Reports? Trends? Interactions?
- Where in our stack?
- Fuzzy deadline
- Yay Fun! Time to get started

FIRST STEPS

- What other resources will become available
- Designer will require strong visual control
- Sales & Marketing need to be be able to **Tell** and then **Show** long before **Sell** and **Use**
- Ideas come from anyone, how can others “get their hands on” my work as I go
- In app or not, delivery will run through me - how do I end in “my specialty”?

FIRST STEPS

- Break down the problem
- Decouple your unknowns to enforce patterns at the boundaries
- Chaos inside the box, order outside



- Make safe tech choices

AT LAST, PYTHON (ETL)

- I ain’t writing Java
- Our backend guys aren’t gonna write rails or javascript
- Python already being used as glue/operations
- I don’t know it (python2 toy project @ university)
- I want to learn it
- Widely used (startups need hiring strategies)
- Actually super good for this use case

JOYS OF SMALL DATA

- If your data fits on a laptop it is small data
- (Also if it fits on a phone)
- (Also if it fits in Chrome's Local Storage)
- Just use an SQL variant
- We had Postgres from Rails/Heroku legacy
- Unknown: Schema (might be circular)

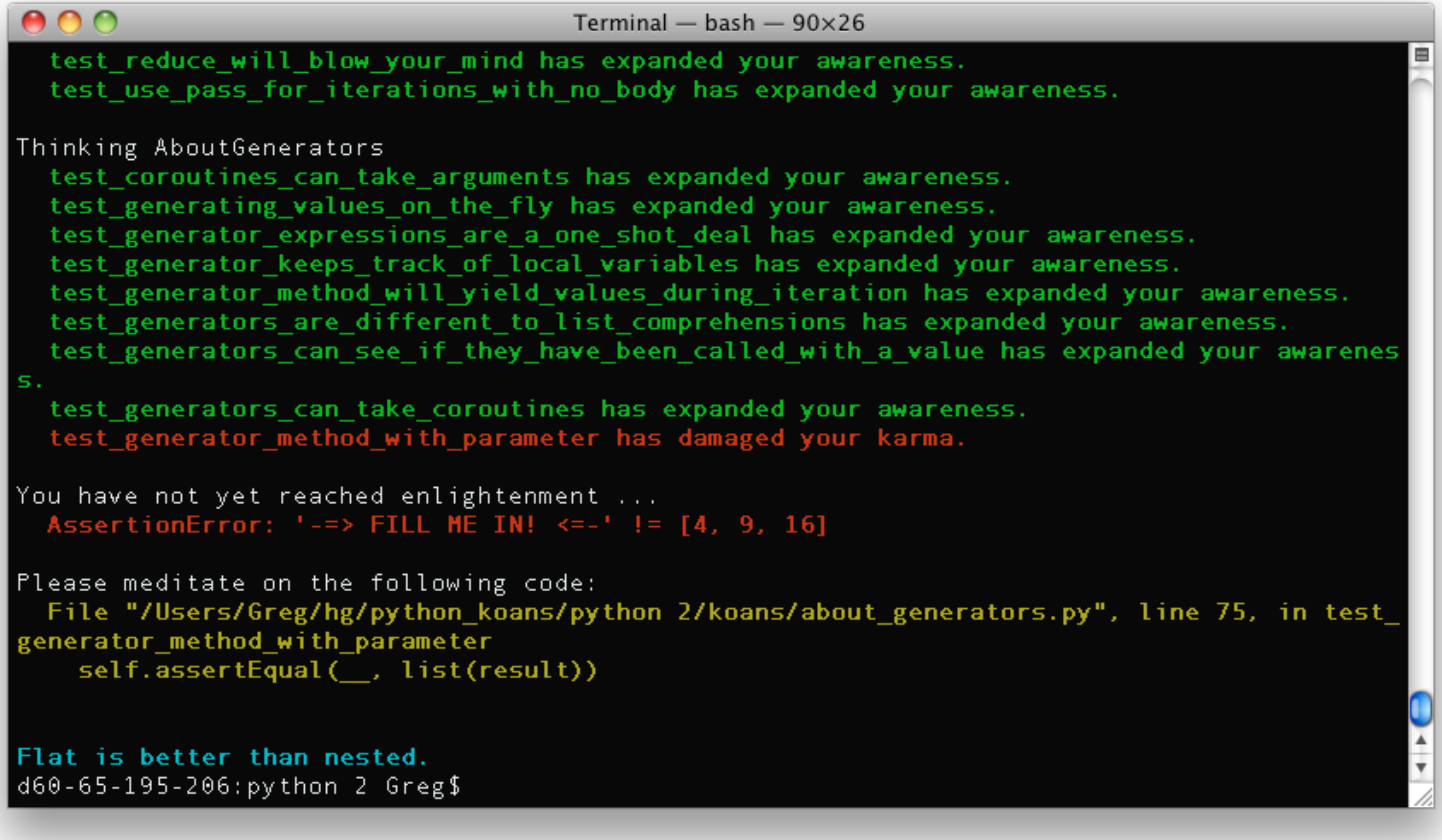
?? VIEW ??

- What & How are unknown
- Why is *sorta unknown*
- *Come back to this one later…*

ASK AROUND

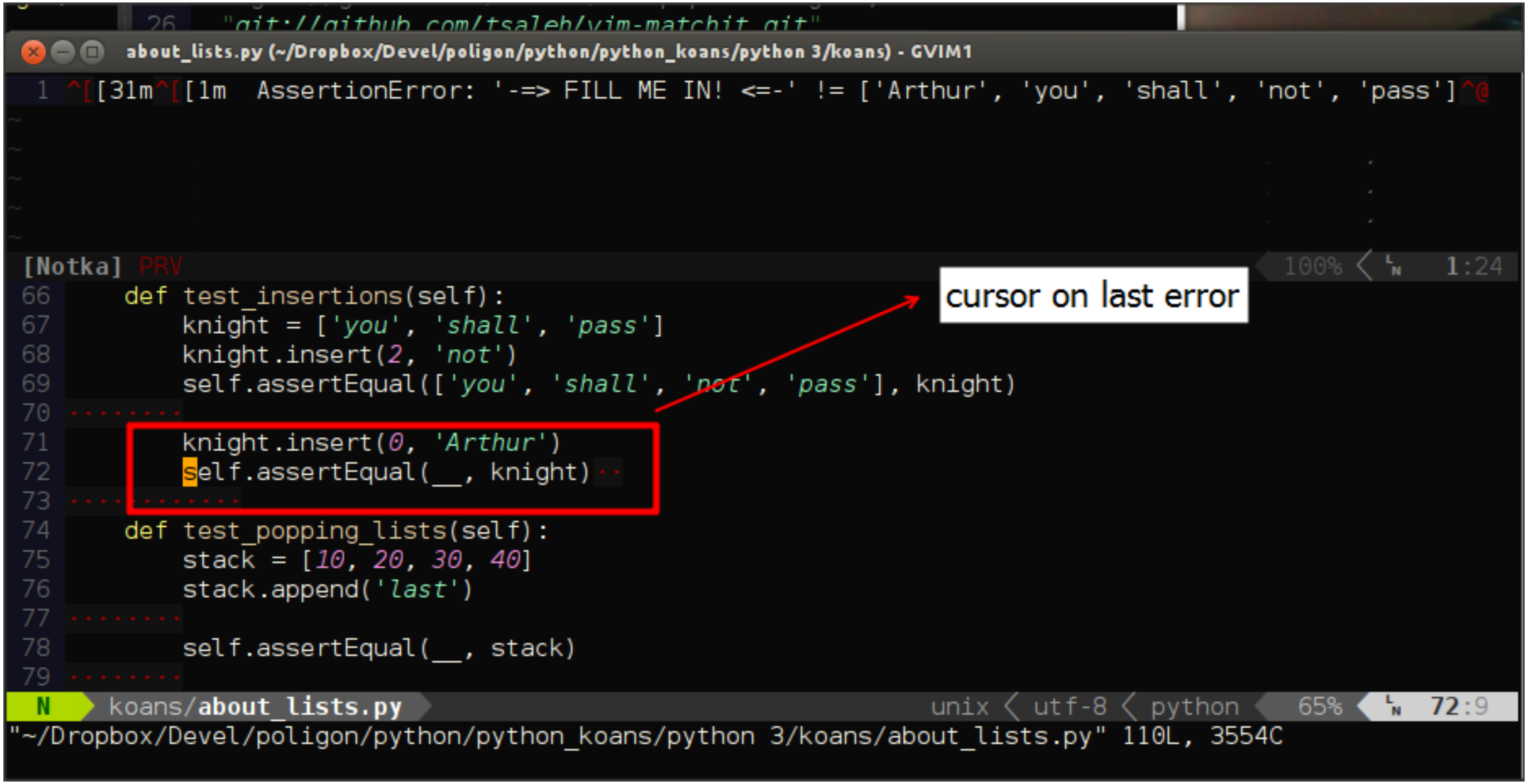
- Python already calls into our other services
- Colleagues know how to use it
- Huge community
- Ask for help!
- Interactive learning

KOANS



```
  test_reduce_will_blow_your_mind has expanded your awareness.
  test_use_pass_for_iterations_with_no_body has expanded your awareness.

Thinking AboutGenerators
  test_coroutines_can_take_arguments has expanded your awareness.
  test_generating_values_on_the_fly has expanded your awareness.
  test_generator_expressions_are_a_one_shot_deal has expanded your awareness.
  test_generator_keeps_track_of_local_variables has expanded your awareness.
  test_generator_method_will_yield_values_during_iteration has expanded your awareness.
  test_generators_are_different_to_list_comprehensions has expanded your awareness.
  test_generators_can_see_if_they_have_been_called_with_a_value has expanded your awarenes
s.
  test_generators_can_take_coroutines has expanded your awareness.
  test_generator_method_with_parameter has damaged your karma.

You have not yet reached enlightenment ...
  AssertionError: '-=> FILL ME IN! <=-' != [4, 9, 16]

Please meditate on the following code:
  File "/Users/Greg/hg/python_koans/python 2/koans/about_generators.py", line 75, in test_
generator_method_with_parameter
    self.assertEqual(__, list(result))


Flat is better than nested.
d60-65-195-206:python 2 Greg$
```

KOANS



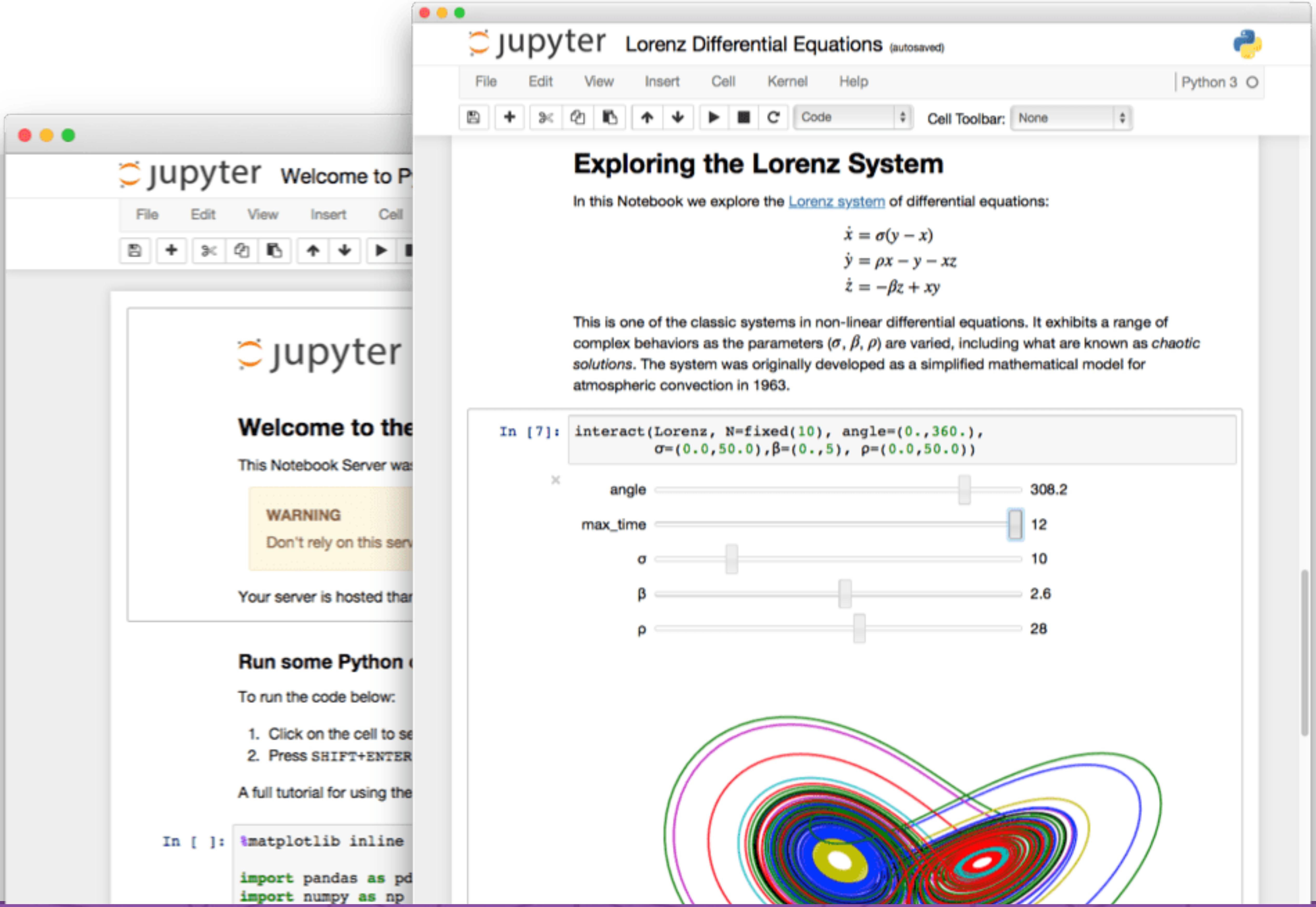
"git://github.com/tsaleh/vim-matchit.git"

about_lists.py (~/Dropbox/Devel/poligon/python/python_koans/python 3/koans) - GVIM1

```
^[[31m^[[1m  AssertionError: '-=> FILL ME IN! <=-' != ['Arthur', 'you', 'shall', 'not', 'pass']^@
~
~
~
~
~
[Notka] PRV                                                         100%  1:24
    def test_insertions(self):
        knight = ['you', 'shall', 'pass']
        knight.insert(2, 'not')
        self.assertEqual(['you', 'shall', 'not', 'pass'], knight)

        knight.insert(0, 'Arthur')
        self.assertEqual(__, knight)

    def test_popping_lists(self):
        stack = [10, 20, 30, 40]
        stack.append('last')

        self.assertEqual(__, stack)

 N  koans/about_lists.py                  unix < utf-8 < python   65%  72:9
"~/Dropbox/Devel/poligon/python/python_koans/python 3/koans/about_lists.py" 110L, 3554C
```

cursor on last error

JUPYTER!



jupyter

Welcome to the

This Notebook Server wa

WARNING

Don't rely on this serv

Your server is hosted thar

Run some Python c

To run the code below:

1. Click on the cell to se
2. Press SHIFT+ENTER

A full tutorial for using the

```
In [ ]: %matplotlib inline

        import pandas as pd
        import numpy as np
```

Cell Toolbar: None

Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                 σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
```

angle 308.2

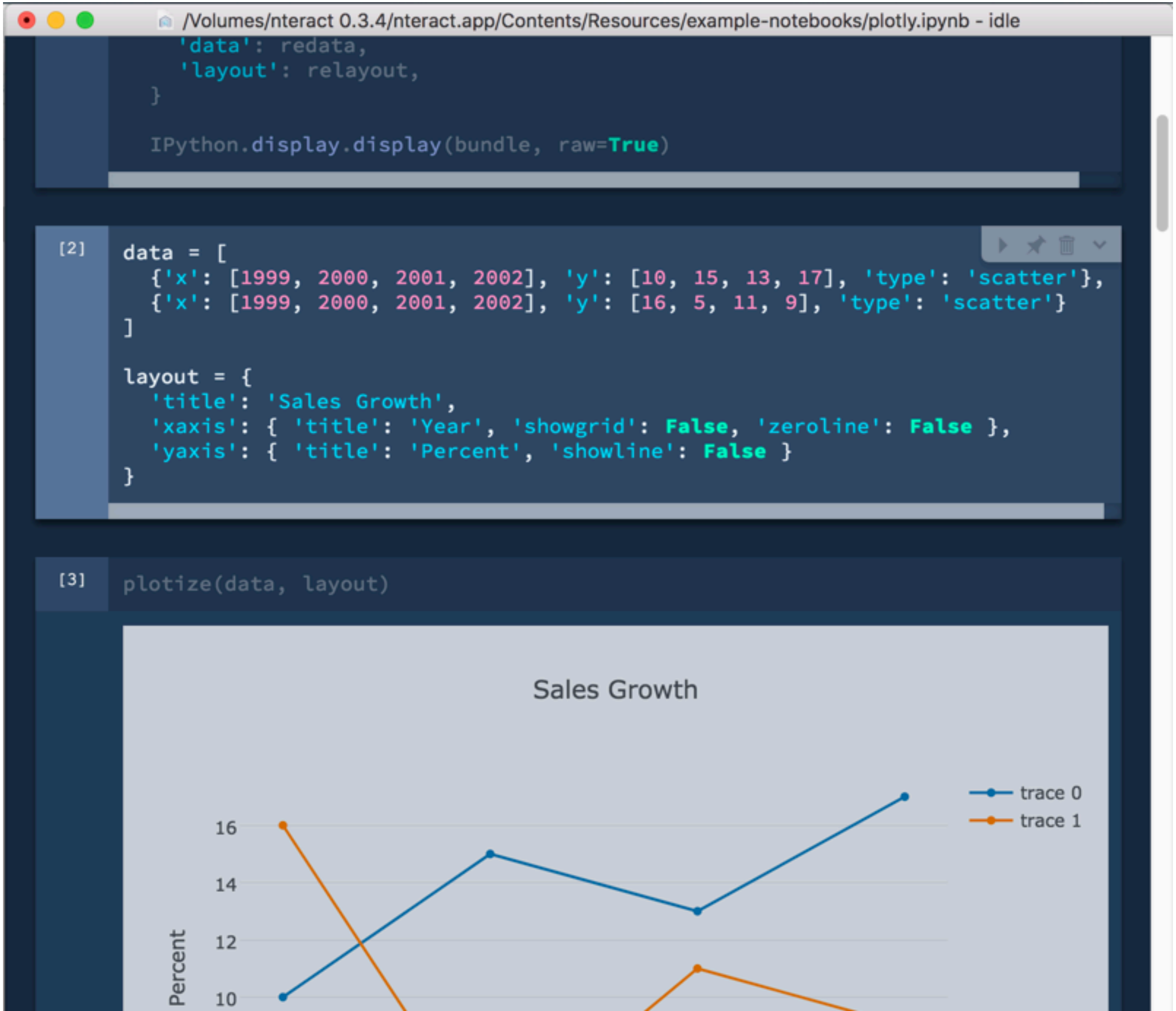
max_time 12

σ 10

β 2.6

ρ 28

NTERACT / HYDROGEN



/Volumes/nteract 0.3.4/nteract.app/Contents/Resources/example-notebooks/plotly.ipynb - idle

```
    'data': redata,
    'layout': relayout,
  }

  IPython.display.display(bundle, raw=True)
```

[2]

```
data = [
  {'x': [1999, 2000, 2001, 2002], 'y': [10, 15, 13, 17], 'type': 'scatter'},
  {'x': [1999, 2000, 2001, 2002], 'y': [16, 5, 11, 9], 'type': 'scatter'}
]

layout = {
  'title': 'Sales Growth',
  'xaxis': { 'title': 'Year', 'showgrid': False, 'zeroline': False },
  'yaxis': { 'title': 'Percent', 'showline': False }
}
```

[3]

```
plotize(data, layout)
```

CODE VOMIT

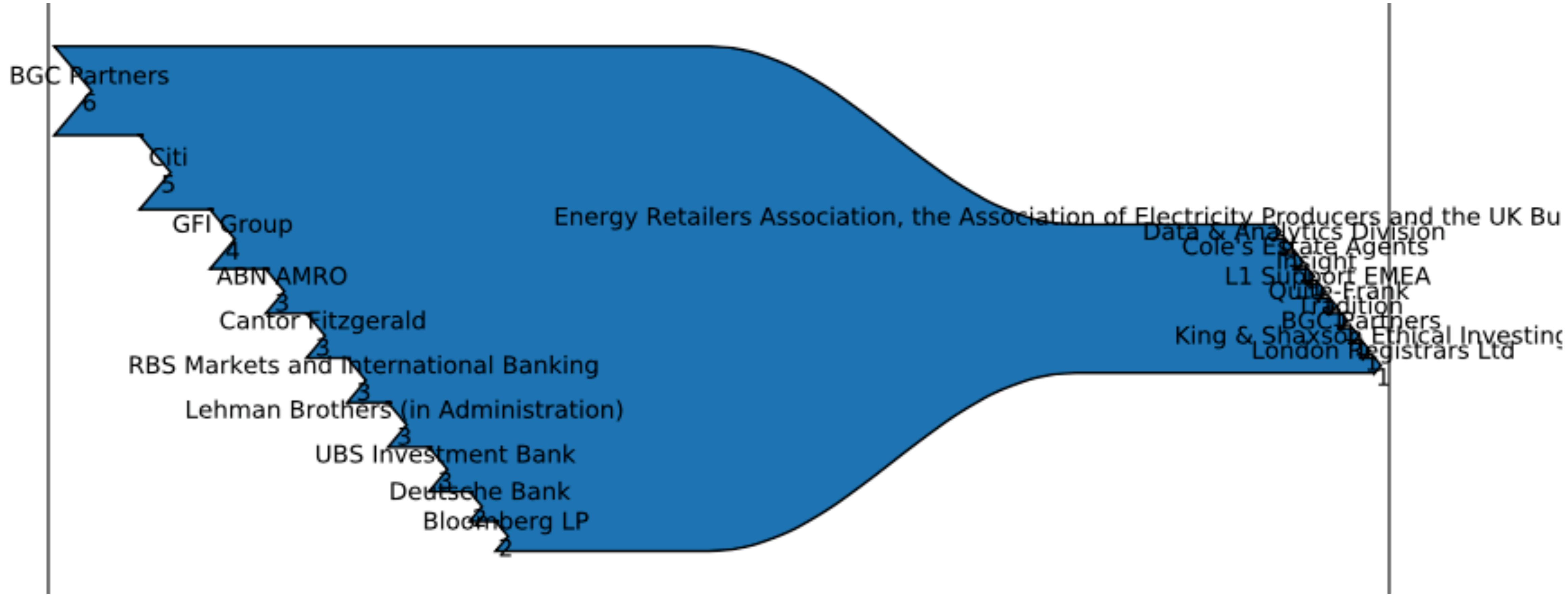
- Coding to an uncertain future
- Break the rules
- Simple code, no magic
- Performance (non) considerations
- Be verbose
- Write for the next human

PANDAS (as DB)

- 🔥 Wes’s Keynote 🔥
- Super fast and easy to use
- Skip defining a schema
- Skip migrations
- Skip having a server
- Dump your data into ~~tables~~ dataframes, have a play

CEO HAS IDEA

- Sources & Destinations
- Deadline is tomorrow
- Matplotlib



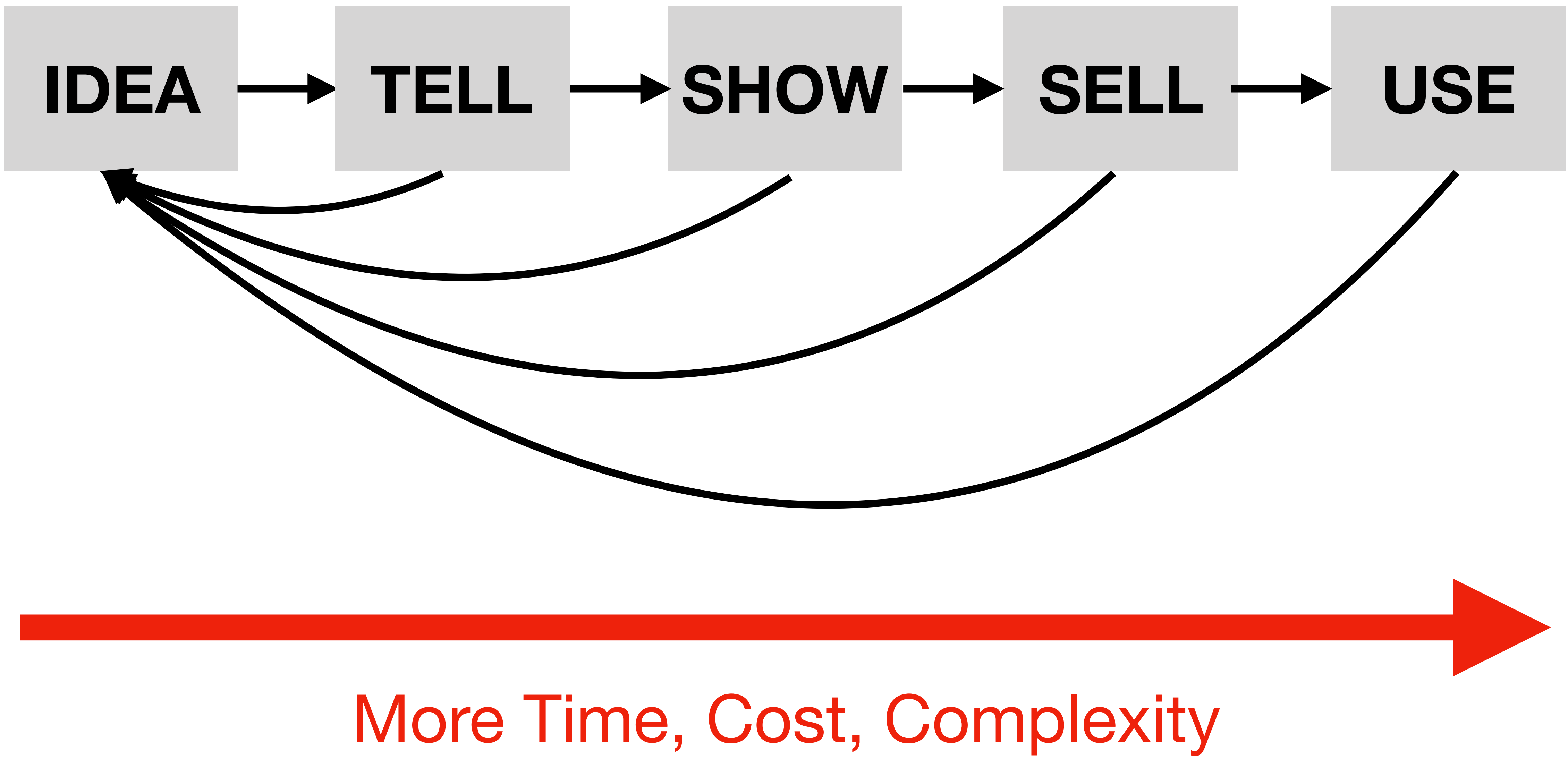
DESIGN

- Huge risk: how will the view look & feel
- Tech decisions affect non-tech team members
- UX learning curve
- API learning curve

INTERNAL USERS

- Identify your stakeholders
- Leadership
- Sales & Marketing **TELL->SHOW** (non tech)
- Design - ability to edit

FEEDBACK LOOPS



IDEA → TELL → SHOW → SELL → USE

More Time, Cost, Complexity

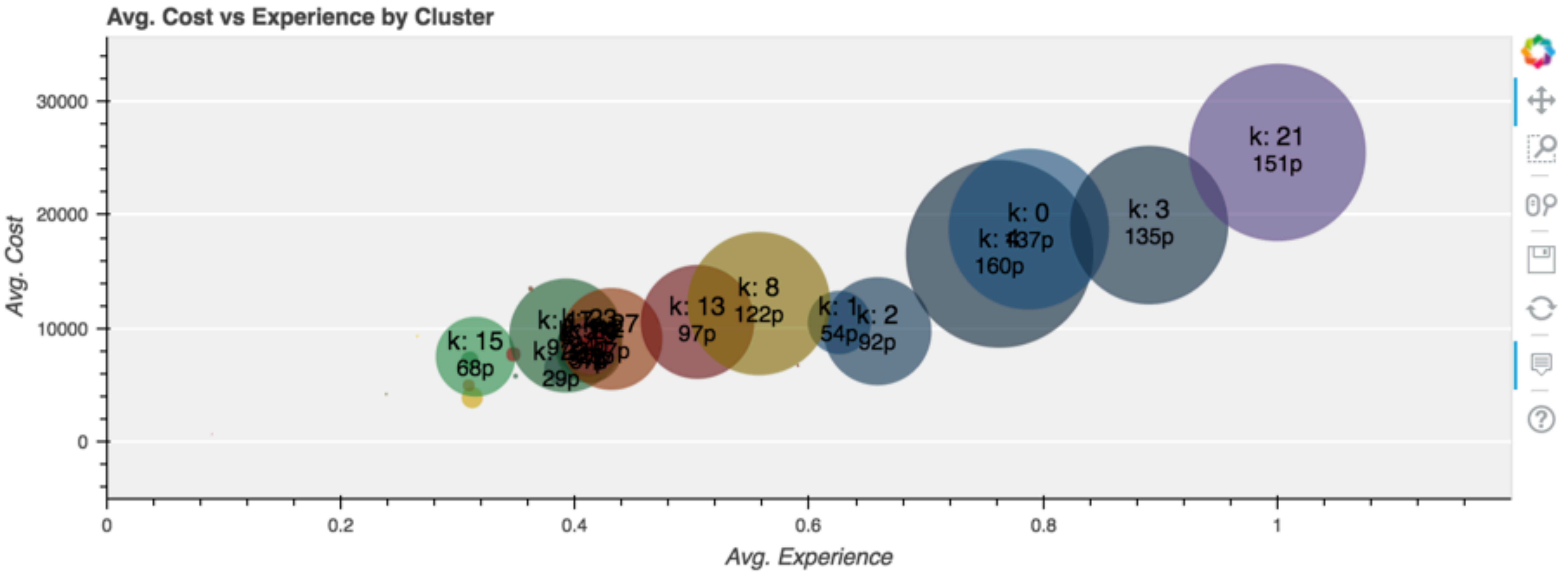
“PRODUCTION”

- “Move fast and break things”
- “If it isn’t in Production, its vaporware”
- Be transparent
- Allow collaboration & iteration
- **Even the biggest achievements roll out as a series of steps**
- MVP: Jupyter
- **But what is the actual production plan?**

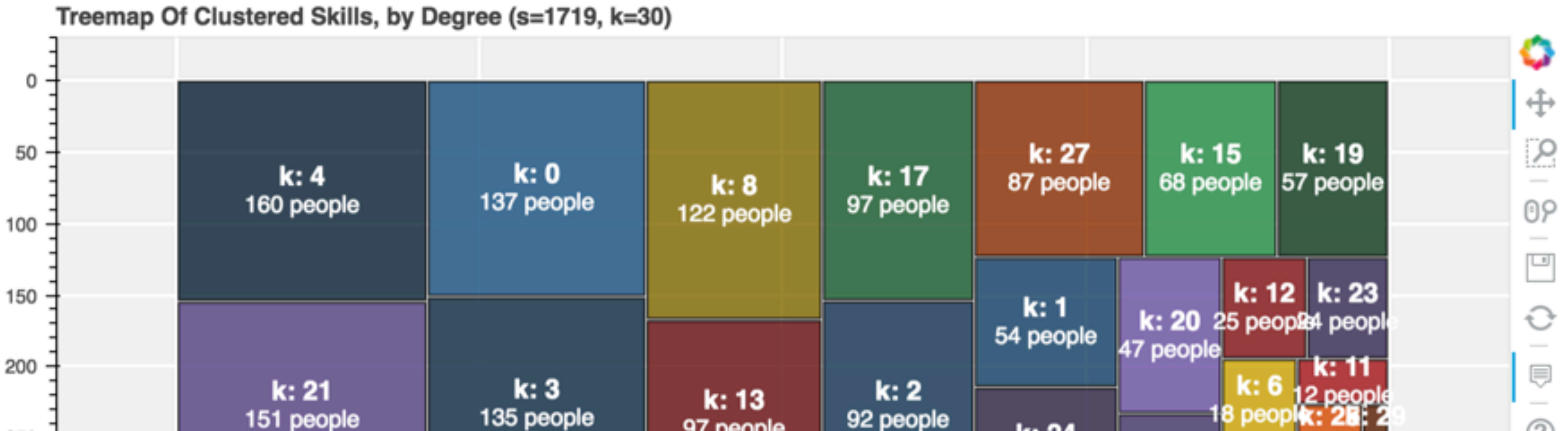
BOKEH

- No offence to matpotlib
- JS developers love D3.js. We love it so much
- Python + D3 + Jupyter? BOKEH!
- (Not actually D3, but philosophically similar)
- Allows export (salesware)
- Allows interaction (demoware)
- Our designer can relax

BOKEH



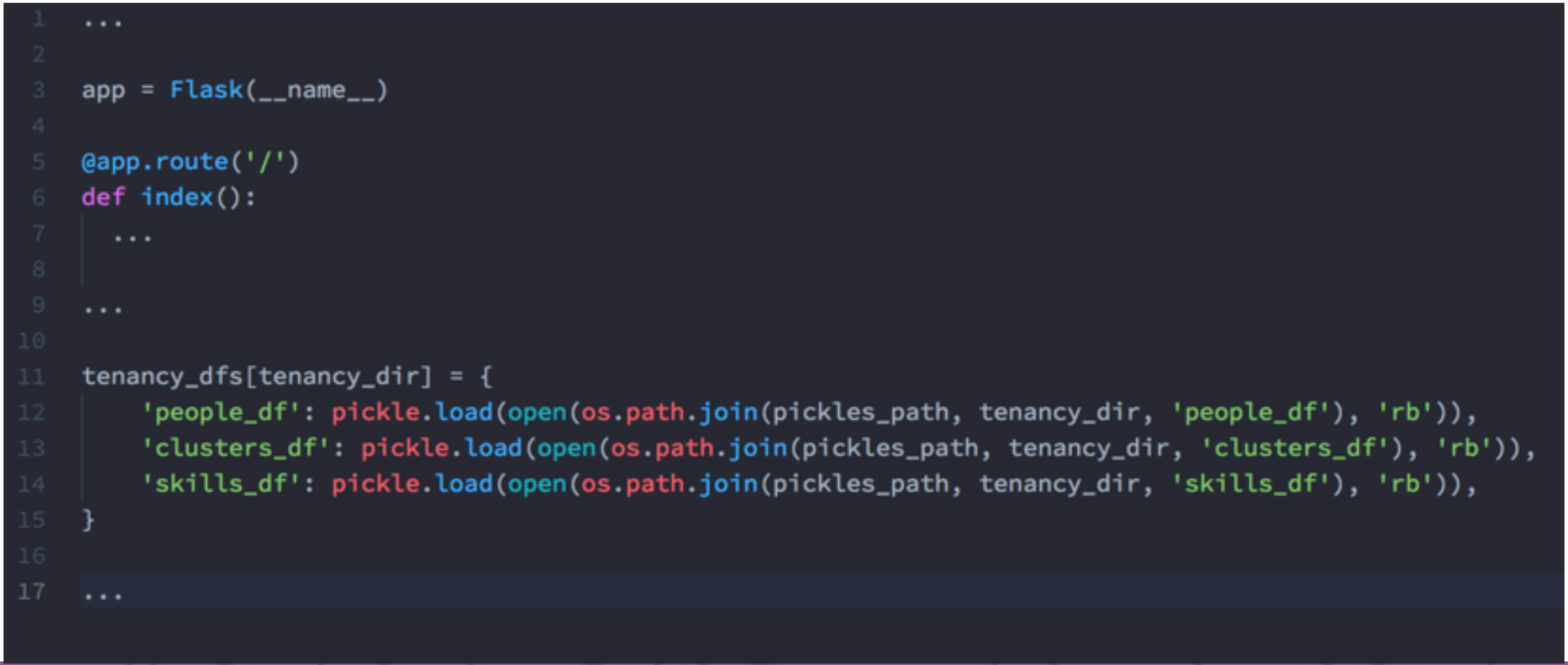
Avg. Cost vs Experience by Cluster



Treemap Of Clustered Skills, by Degree (s=1719, k=30)

HEROKU

- git push master === production
- Add some flask
- Load your pandas dataframes at boot (via pickle)

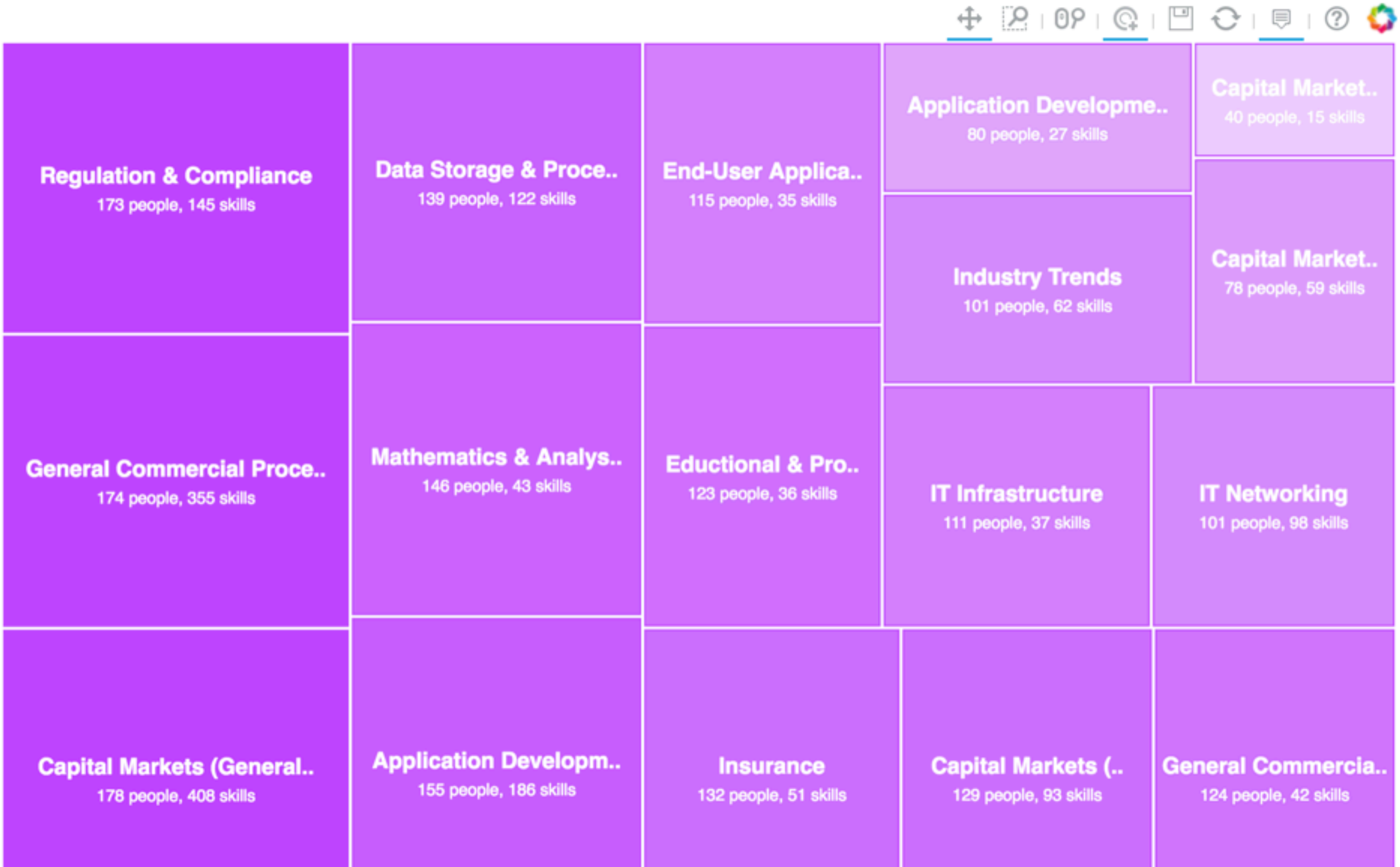


```
...

app = Flask(__name__)

@app.route('/')
def index():
  ...

...

tenancy_dfs[tenancy_dir] = {
    'people_df': pickle.load(open(os.path.join(pickles_path, tenancy_dir, 'people_df'), 'rb')),
    'clusters_df': pickle.load(open(os.path.join(pickles_path, tenancy_dir, 'clusters_df'), 'rb')),
    'skills_df': pickle.load(open(os.path.join(pickles_path, tenancy_dir, 'skills_df'), 'rb')),
}

...
```

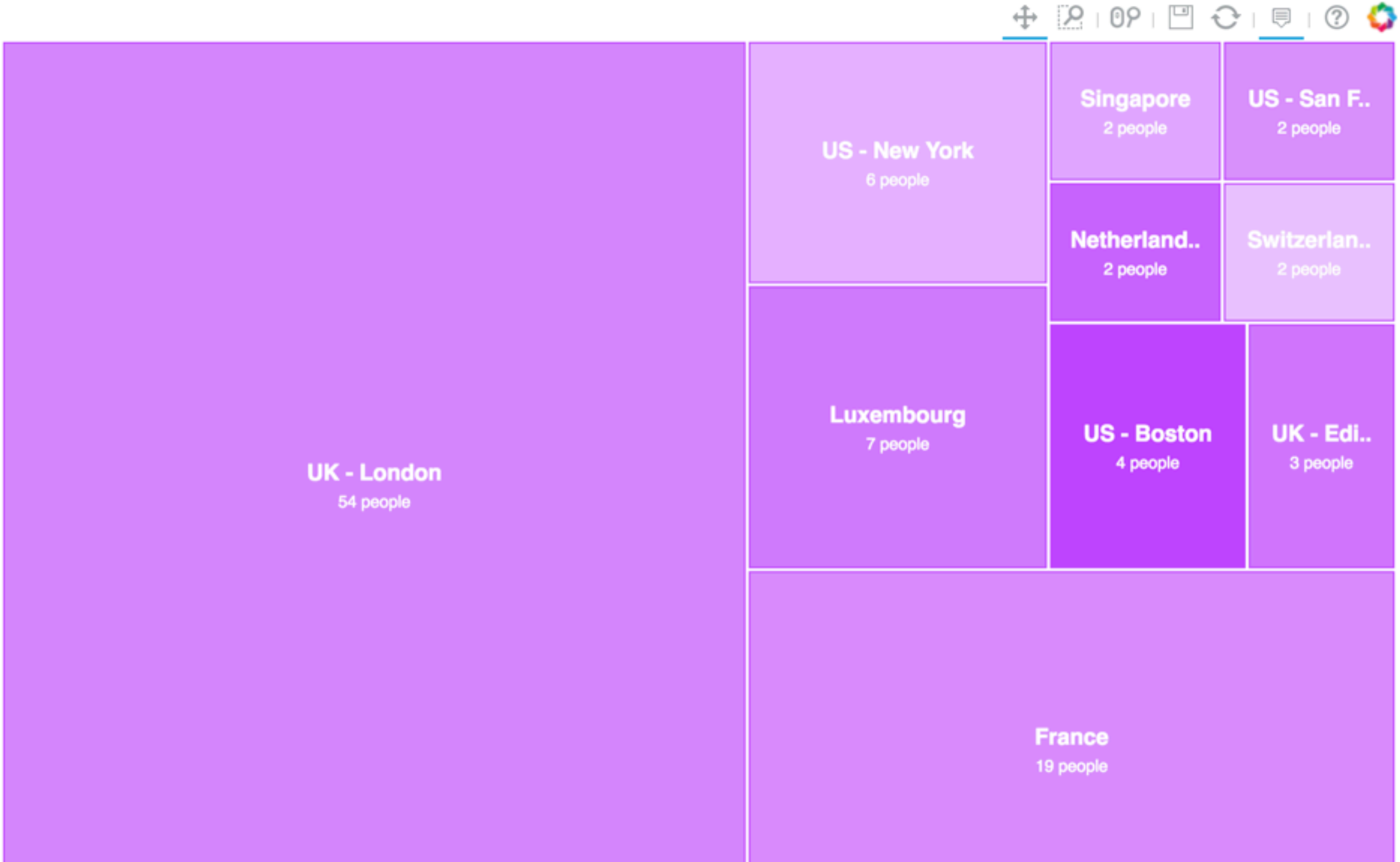
TELL A STORY

- Flask routes = bokeh pages

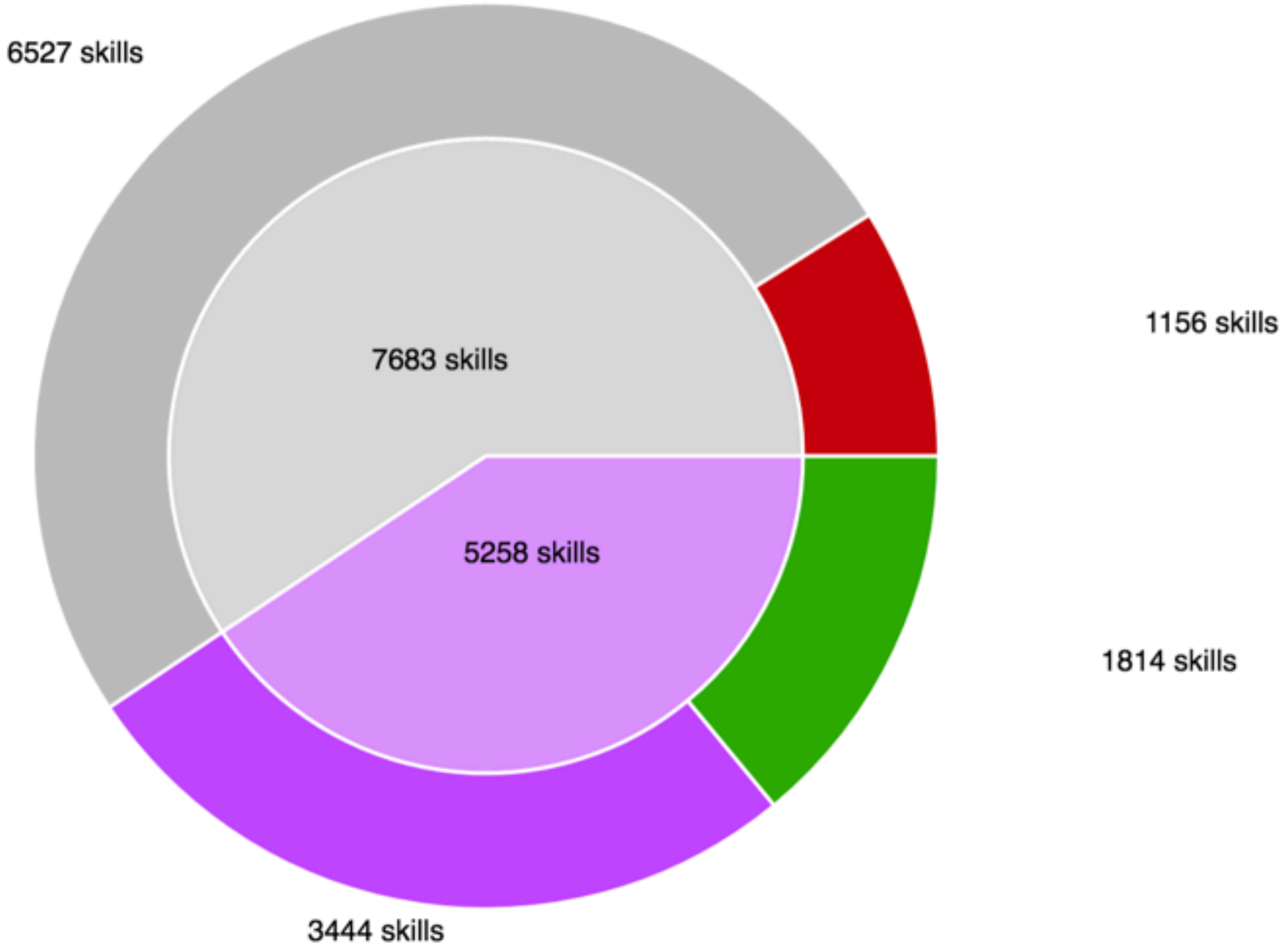
TELL A STORY


Regulation & Compliance
173 people, 145 skills
Data Storage & Proce..
139 people, 122 skills
End-User Applica..
115 people, 35 skills
Application Developme..
80 people, 27 skills
Capital Market..
40 people, 15 skills
Industry Trends
101 people, 62 skills
Capital Market..
78 people, 59 skills
General Commercial Proce..
174 people, 355 skills
Mathematics & Analys..
146 people, 43 skills
Eductional & Pro..
123 people, 36 skills
IT Infrastructure
111 people, 37 skills
IT Networking
101 people, 98 skills
Capital Markets (General..
178 people, 408 skills
Application Developm..
155 people, 186 skills
Insurance
132 people, 51 skills
Capital Markets (..
129 people, 93 skills
General Commercia..
124 people, 42 skills

TELL A STORY


UK - London
54 people
US - New York
6 people
Singapore
2 people
US - San F..
2 people
Netherland..
2 people
Switzerlan..
2 people
Luxembourg
7 people
US - Boston
4 people
UK - Edi..
3 people
France
19 people

TELL A STORY


6527 skills
7683 skills
1156 skills
5258 skills
1814 skills
3444 skills

GOGO POSTGRES

- Far enough to spend time on “proper workflow”
- Pandas dataframes map to tables
- Skip migrations!
- Truncate, create, populate
- Postgres is now a Data API

BI TOOLS

- Qlik, Tableau, Looker, MS Power BI
- Allow Sales/Marketing to draw charts
- Definitely a time trap, didn’t end up being used
- Easy to get a demo
- Licensing and cost challenges for our model
- Another platform to support
- A BI tool is not a product solution
- Super slick to integrate with and get simple charts out

FIRST STORY

- People on the bench are a cost without revenue
- Risk of them leaving the company
- Prediction problem
- Hard to build in BI tools
- Needed a new table
- Small change all along the ETL->PG->View stacks
- Proved the pattern!

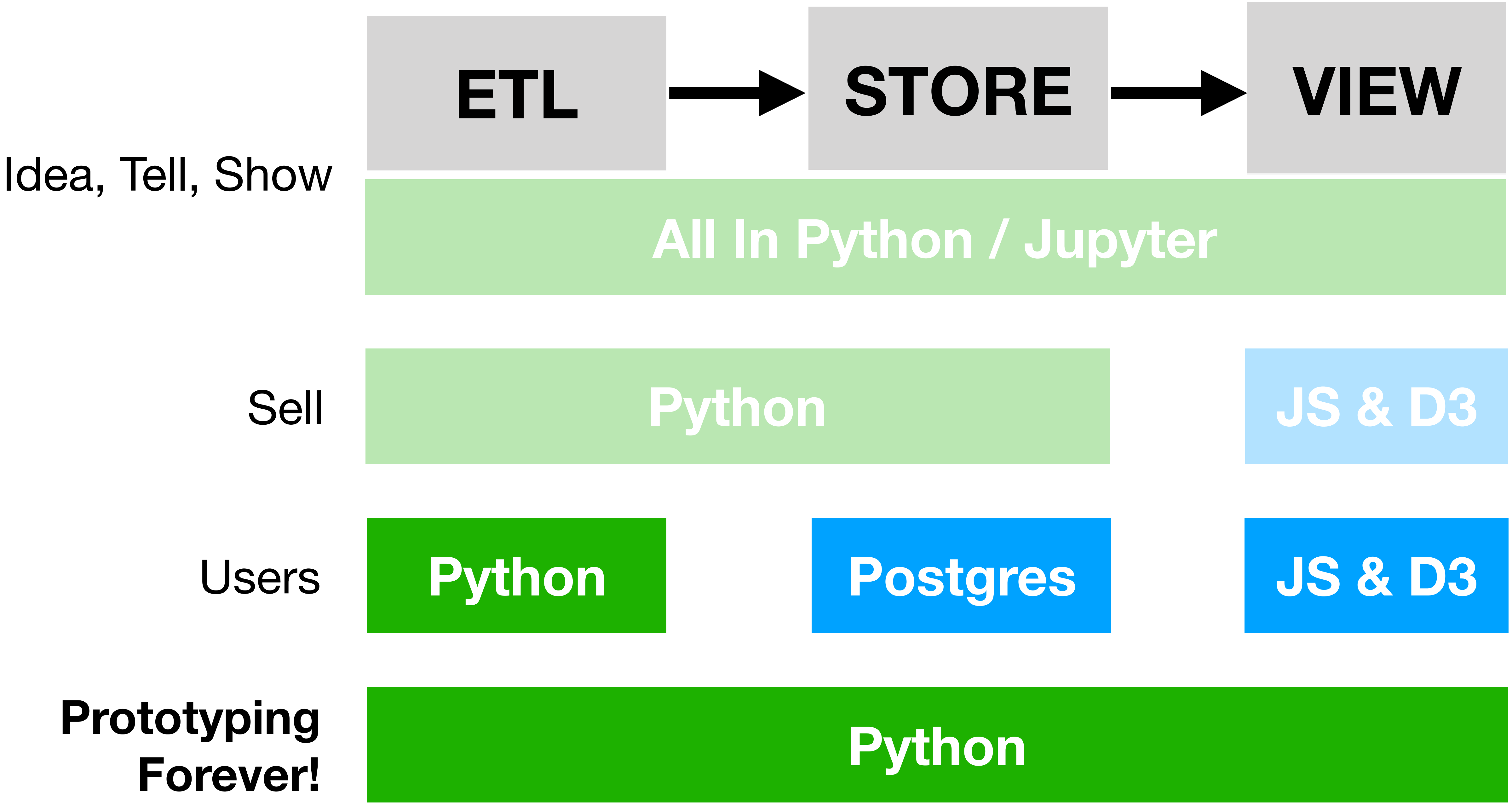
JS WINS IN THE END

- Time to put it in our user web application
- Can keep everything but the “production” visualisations
- Bokeh → D3.js
- Still using Bokeh to prototype locally
- Chaos buzzwords (ask me later): React, Redux, ES6, Node, socket.io, D3.js, VictoryCharts, SASS…

STILL ETL IN JUPYTER

- Feature is in production, with users, in a web app
- Jupyter still the ETL script
- Infinity frontend work appears, no time for handover
- Jupyter to the rescue!
- Easy for someone else to understand and port
- Now exists as a docopt python script run with cron
- Is a library / has tests and all that “proper” stuff
- I was barely involved!

SOLVED: 3 LAYERS



	ETL	STORE	VIEW
Idea, Tell, Show	All In Python / Jupyter		
Sell	Python		JS & D3
Users	Python	Postgres	JS & D3
Prototyping Forever!	Python		

WHAT DID WE LEARN?

BUILD SIMPLE THINGS

OFF THE SHELF, PLUG THINGS TOGETHER

CODE FOR HUMANS

SILO THE STEPS SO THERE CAN BE CHAOS

PRODUCTION STORY
EARLY
ITERATE

EMPOWER YOUR TEAM

NEW TO YOU

NOT NEW TO OTHERS

GET HELP

END UP IN YOUR DOMAIN

FEATURES ARE ROADMAPS

BYE

- rwasher.com
- github.com/rwasher
- twitter.com/richardwasher
- richard@stitched.io