Data testing and documentation with Great Expectations

Budapest Data Forum - Sept 2020

great_expectations

@expectGreatData

About me (Abe)

- Data scientist/engineer
- Tech-first and “enterprise”
- Human-scale, ethical data

Outline

1. What is pipeline debt?
2. How does Great Expectations beat pipeline debt?
3. The law of the stale wiki

great_expectations

? #1

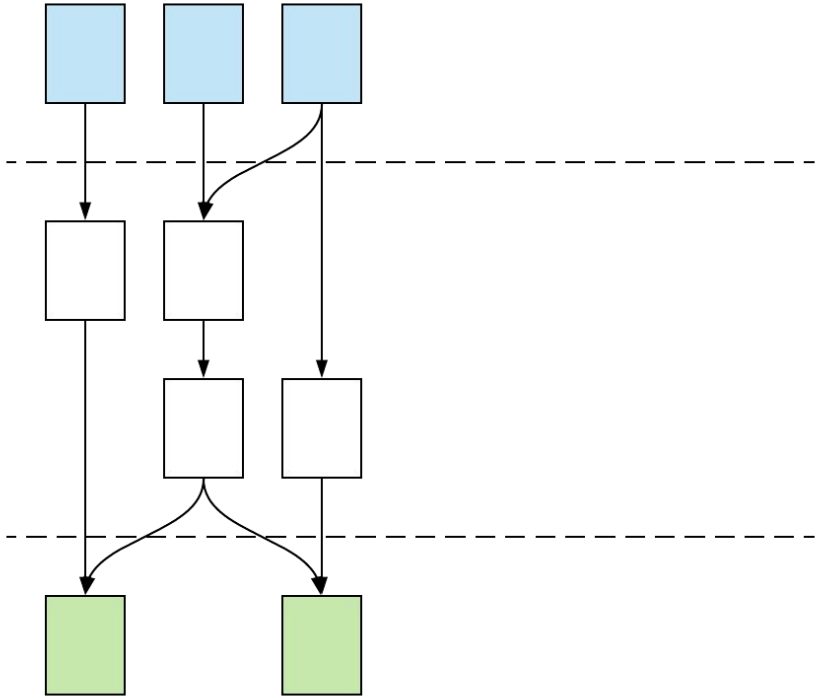
great_expectations

@expectGreatData

What is pipeline debt?

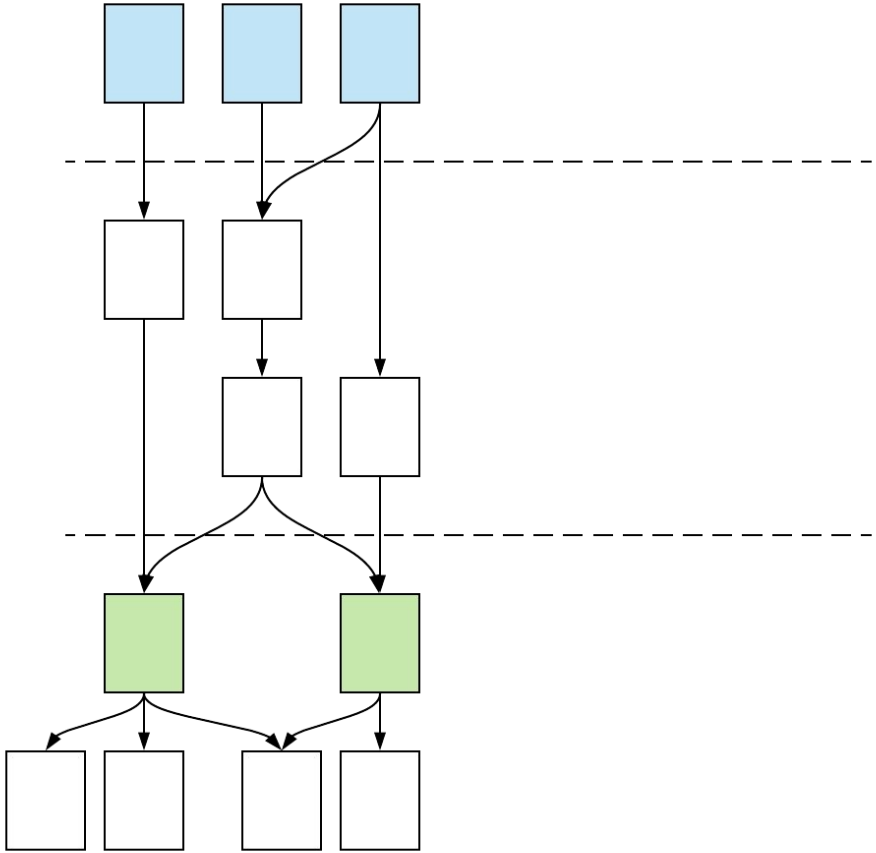
great_expectations

@expectGreatData



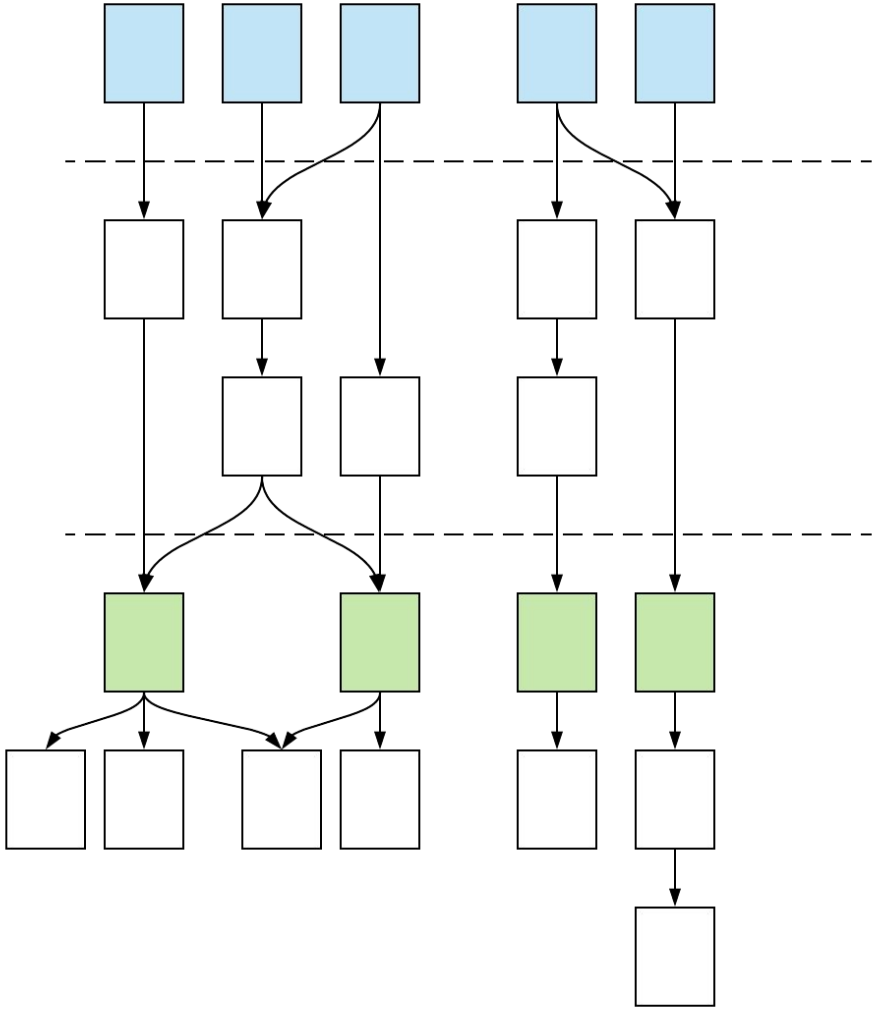
great_expectations

@expectGreatData



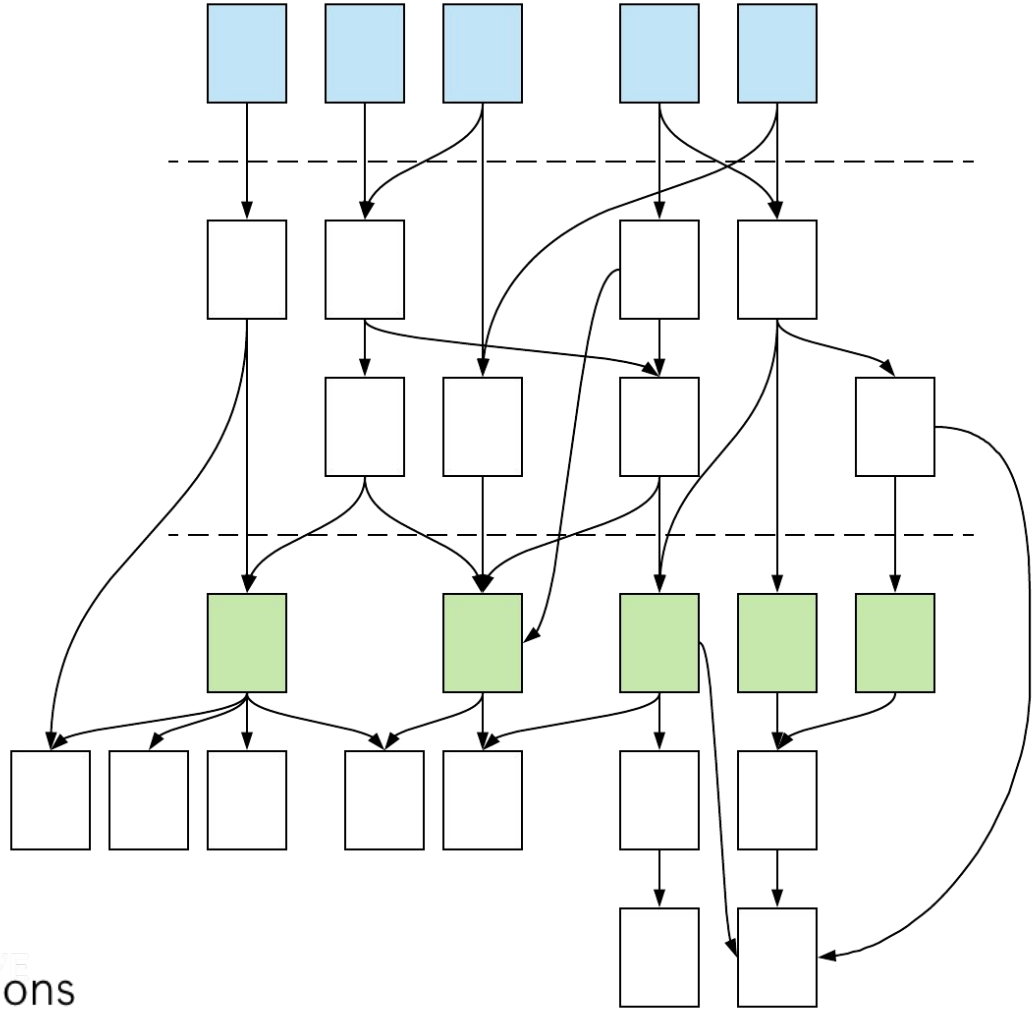
great_expectations

@expectGreatData



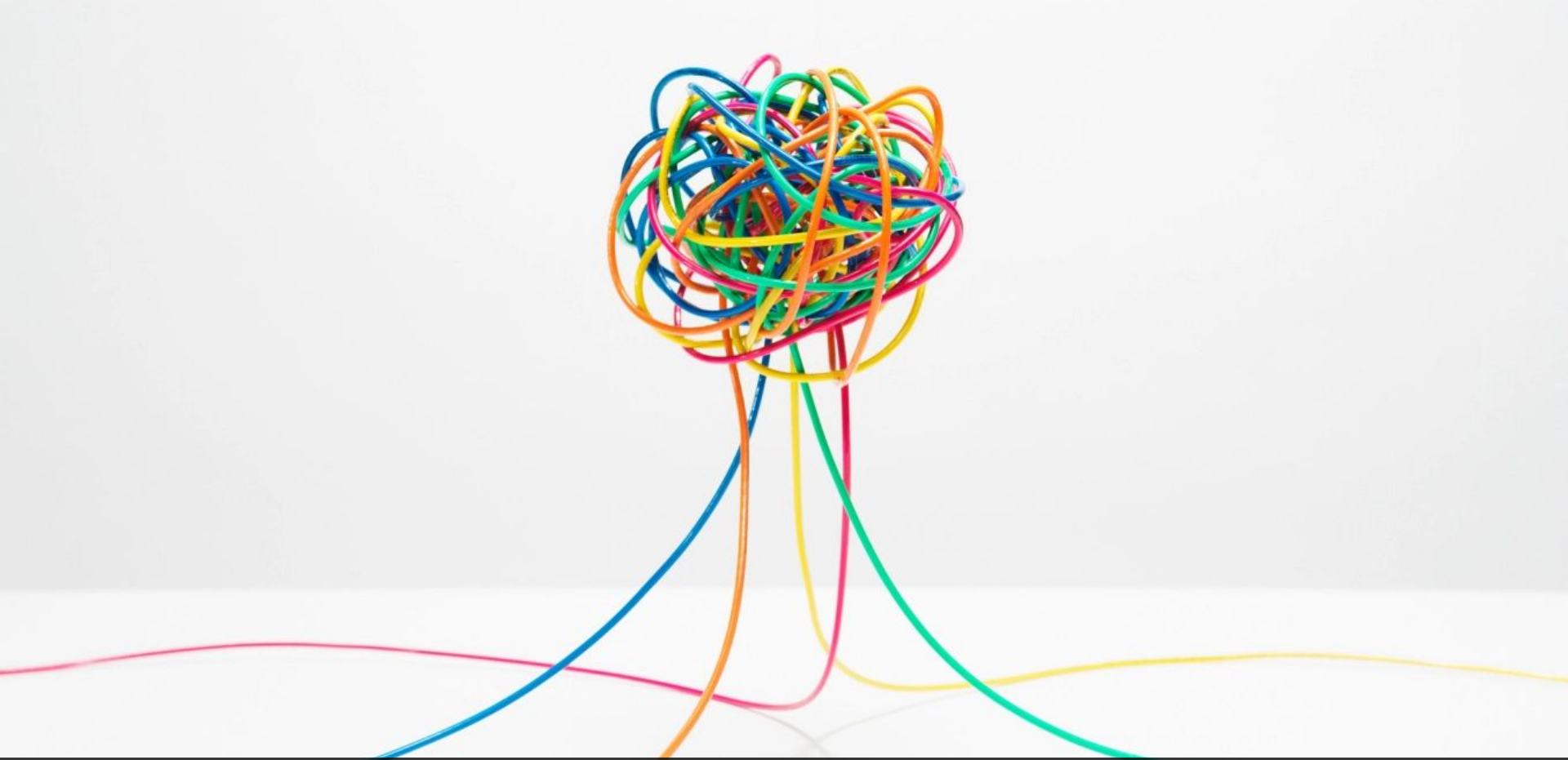
great_expectations

@expectGreatData

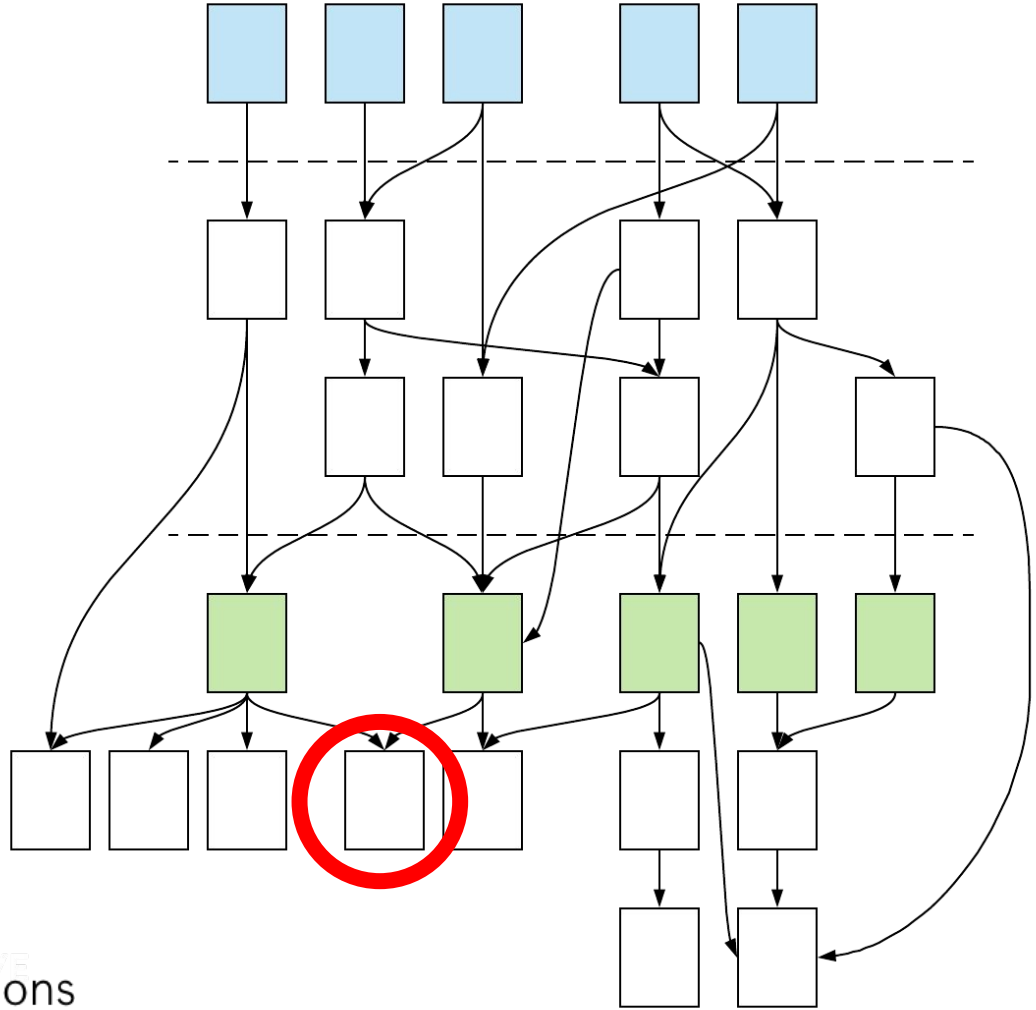


great_expectations

@expectGreatData

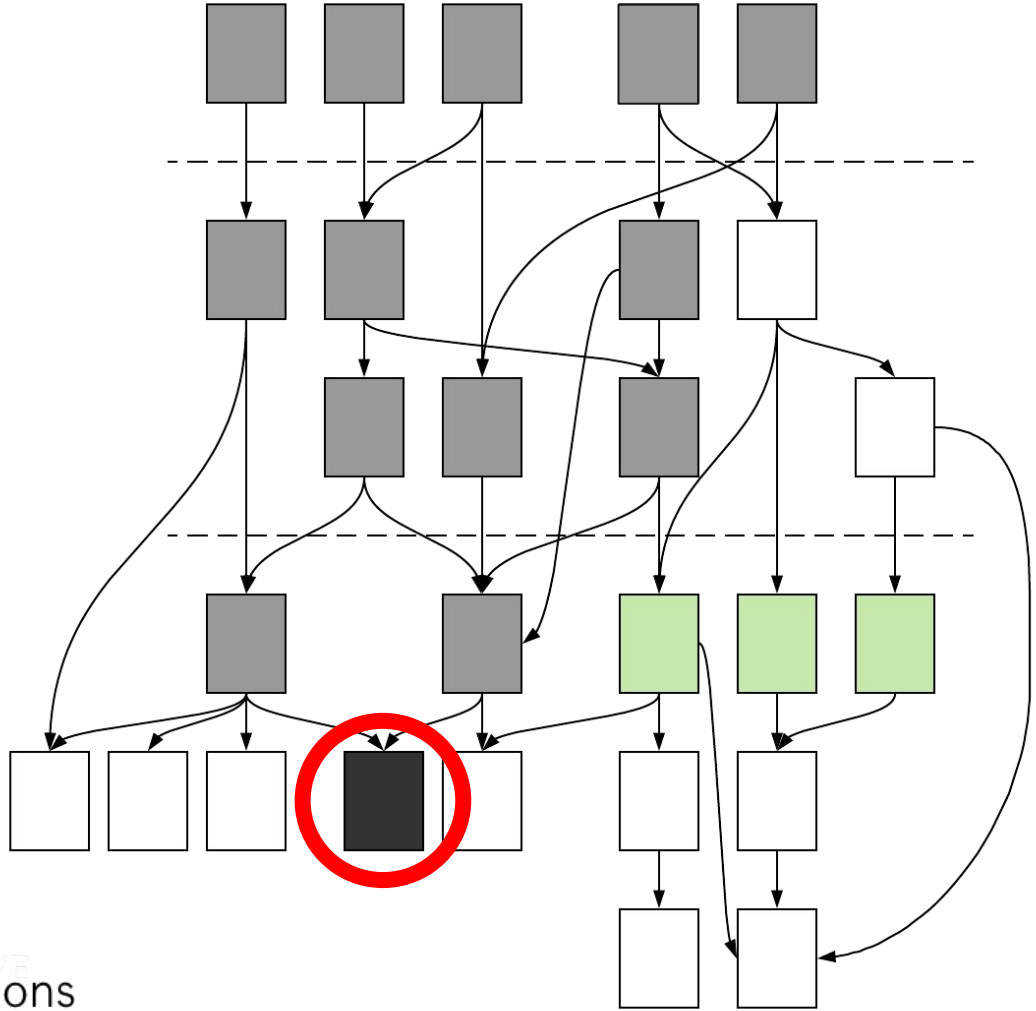


Your data pipelines *want* to be a hairball.



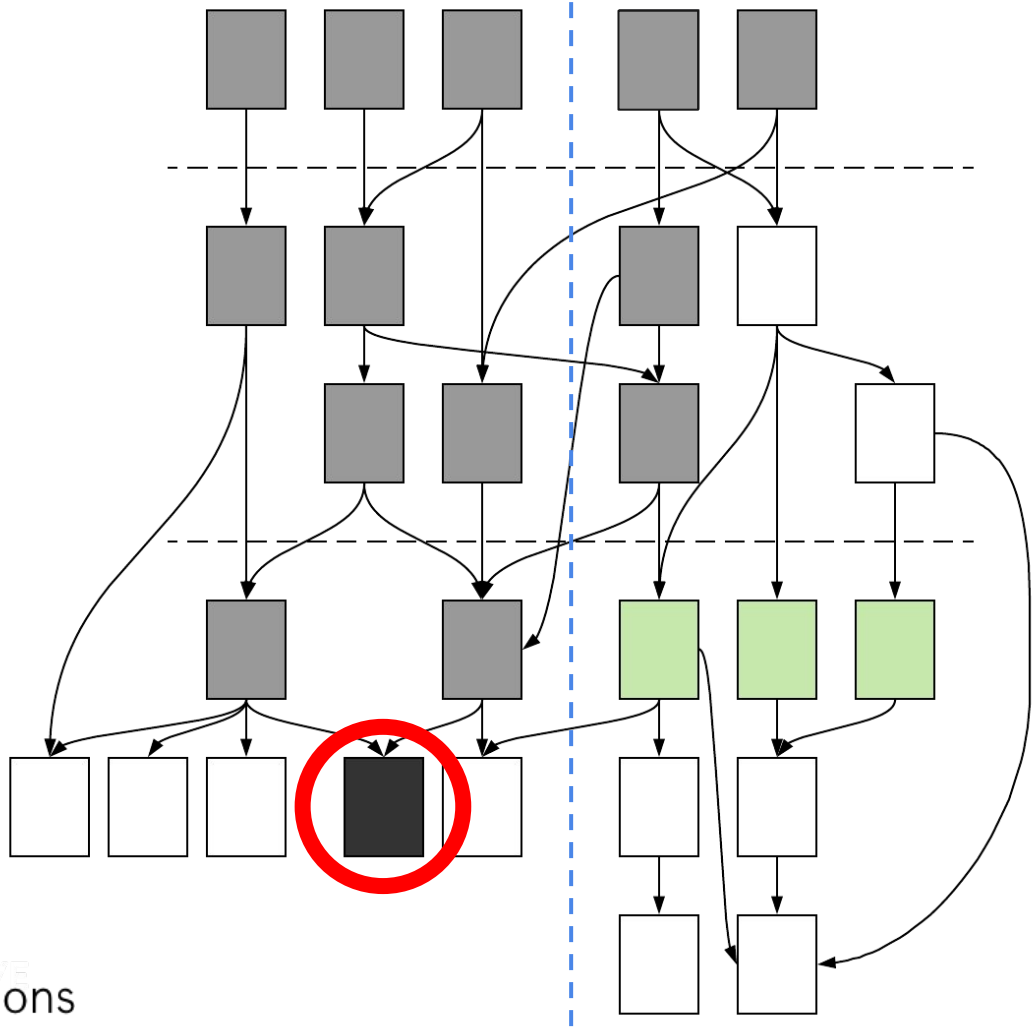
great_expectations

@expectGreatData



great_expectations

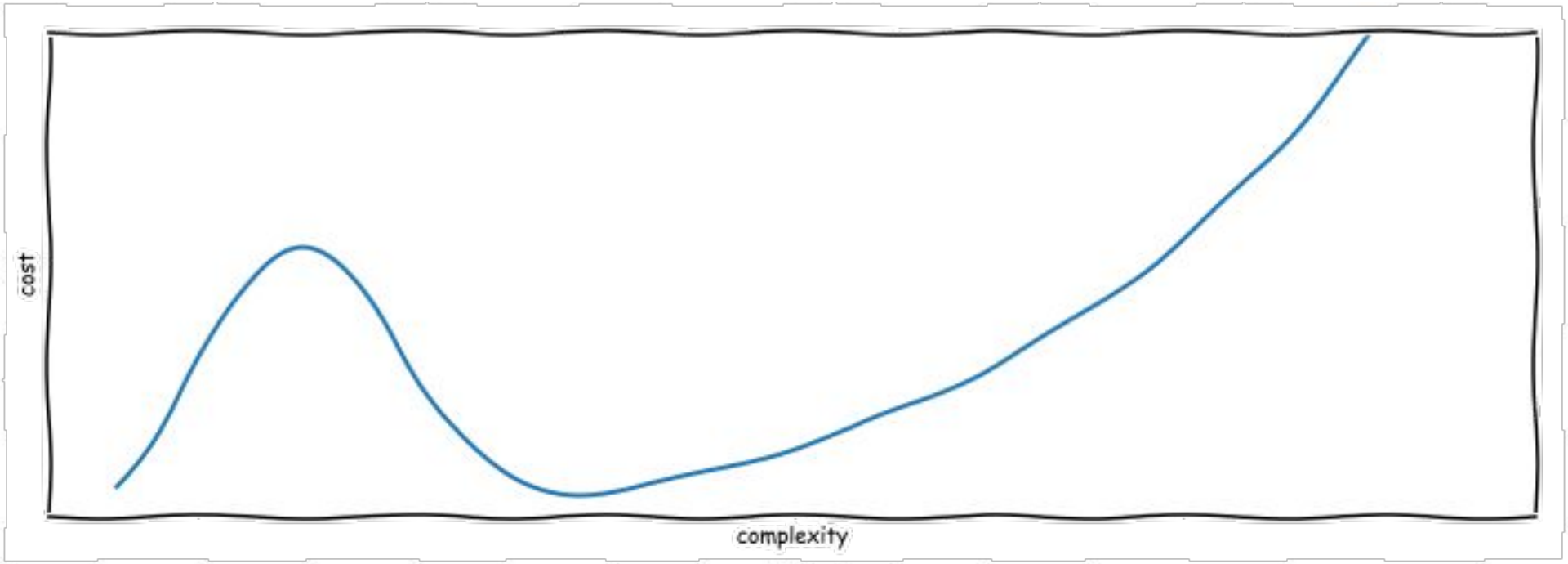
@expectGreatData



great_expectations

@expectGreatData

Creeping, exponential maintenance burden



great_expectations

@expectGreatData

What is pipeline debt?

Technical debt in data pipelines, mainly as a result of missing tests and documentation.

great_expectations

? #2

great_expectations

@expectGreatData

How does Great Expectations beat pipeline debt?

great_expectations

great_expectations

Always know what to expect from your data

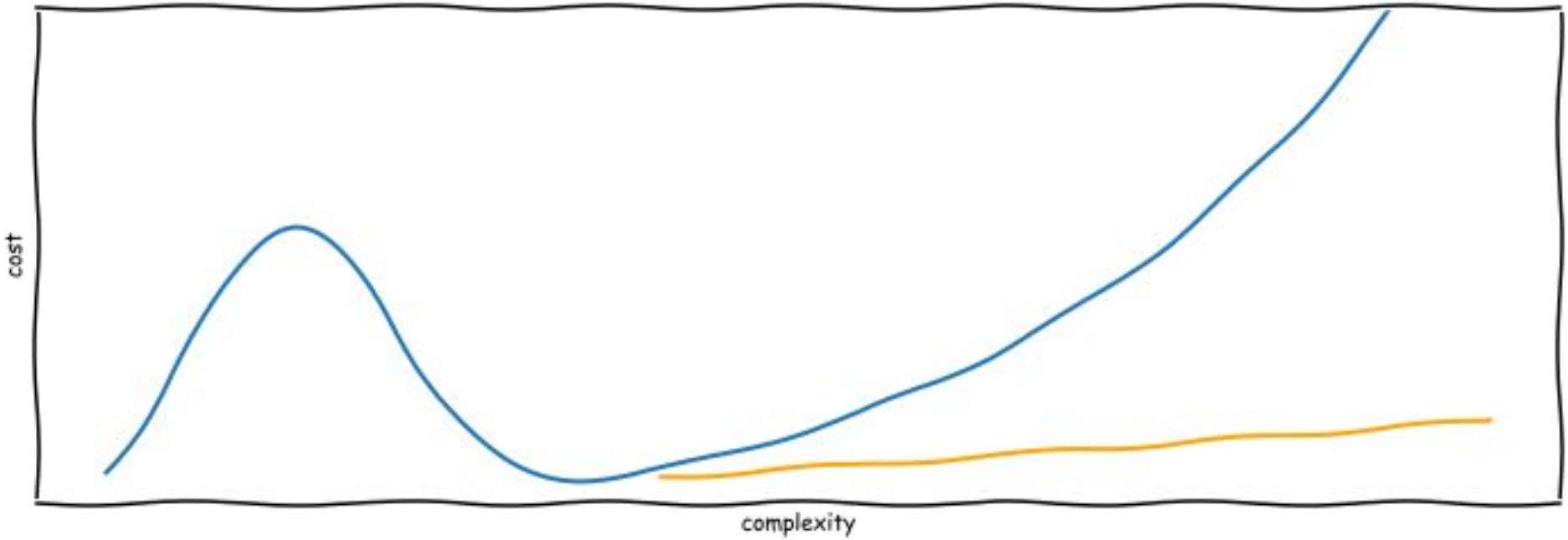
Expectations are assertions about data

great_expectations

```
expect_column_to_exist
expect_table_row_count_to_be_between
expect_column_values_to_be_unique
expect_column_values_to_not_be_null
expect_column_values_to_be_between
expect_column_values_to_match_regex
expect_column_values_to_match_strftime_format
expect_column_mean_to_be_between
expect_column_kl_divergence_to_be_less_than
etc. etc. etc.
```

great_expectations

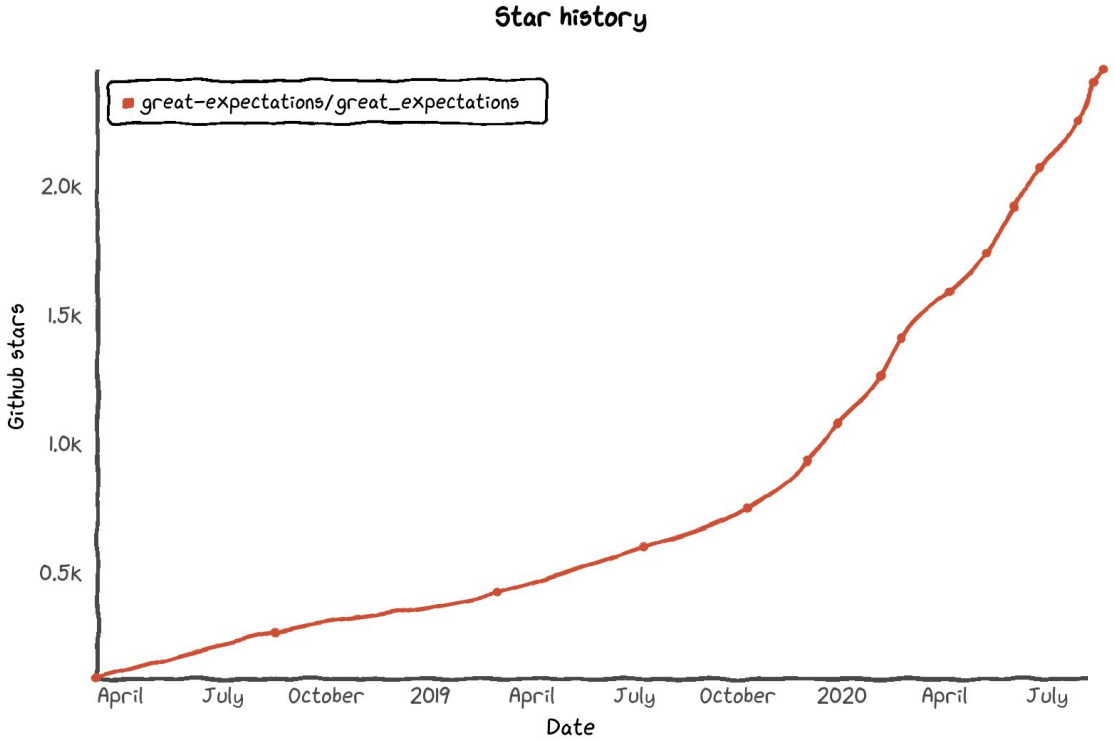
Tests bend the maintenance curve

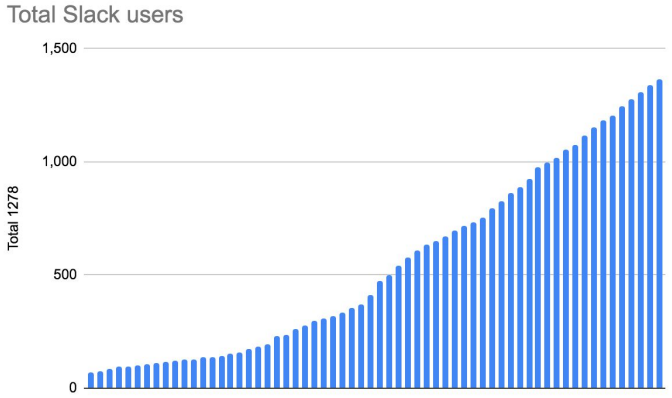


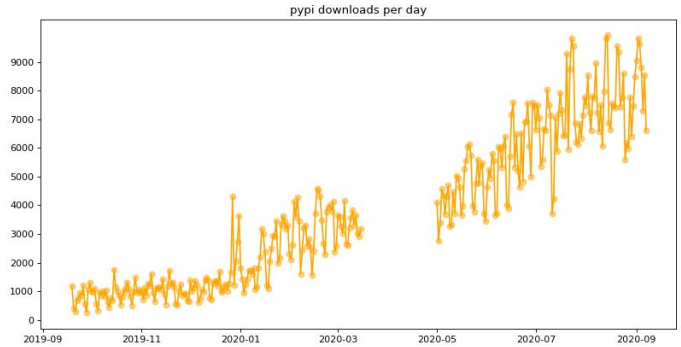
great_expectations

@expectGreatData

Community growth







great_expectations

@expectGreatData

How does Great Expectations beat pipeline debt?

Tests bend the maintenance curve,
making it sublinear instead of superlinear.

great_expectations

@expectGreatData

? #3

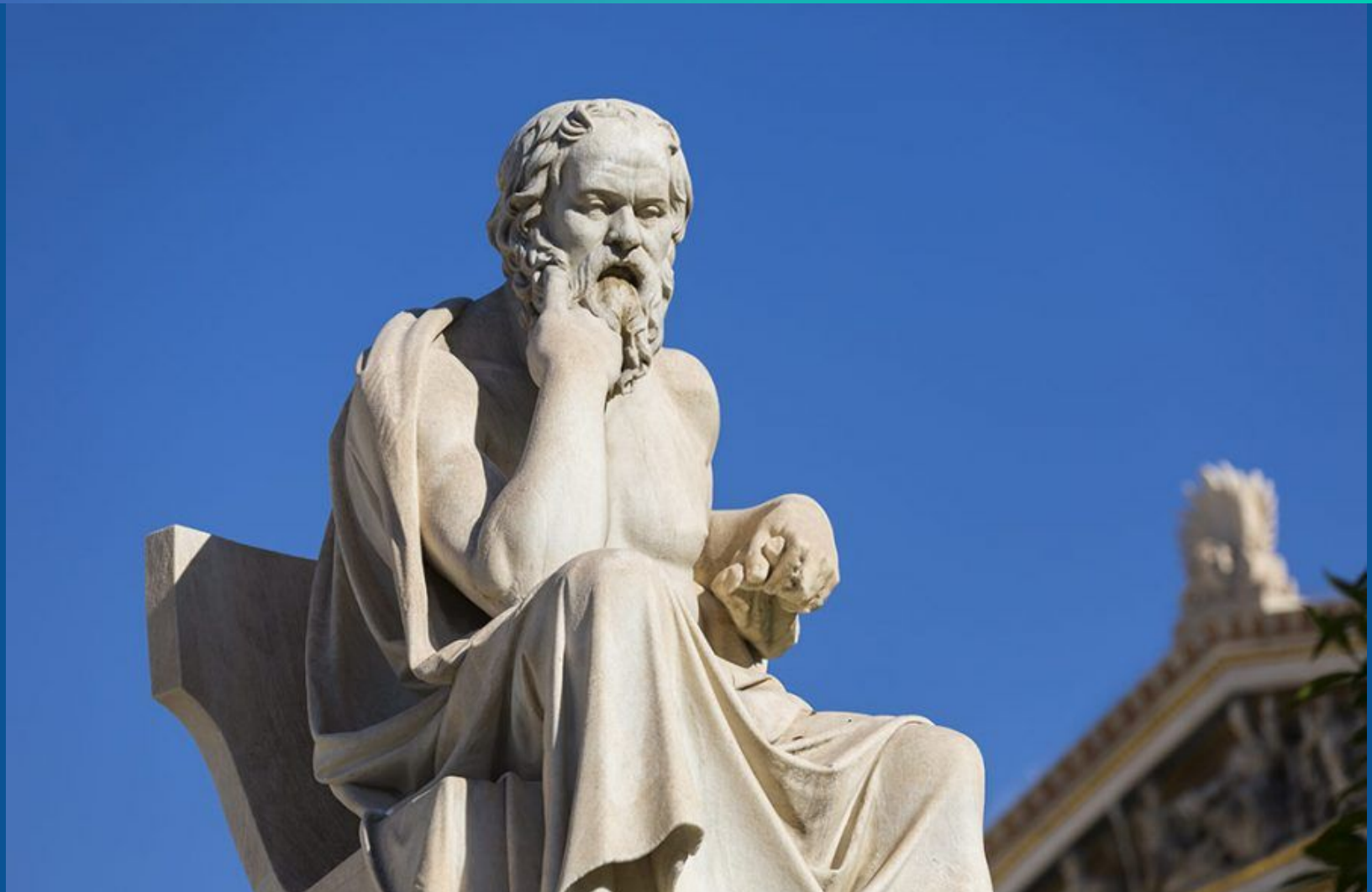
great_expectations

@expectGreatData

The Law of the Stale Wiki

great_expectations

@expectGreatData



Why do we write docs anyway?

Because someone might want to read them later.

Who, specifically, might want to read them? When?

- Your future self. In the future.
- Teammates, when onboarding or coming up to speed in a new area--and just to remember or double check things.
- Other teams, to understand where data comes from, how it should behave, and what they can do with it.

When is cost(writing) < benefit(reading)?

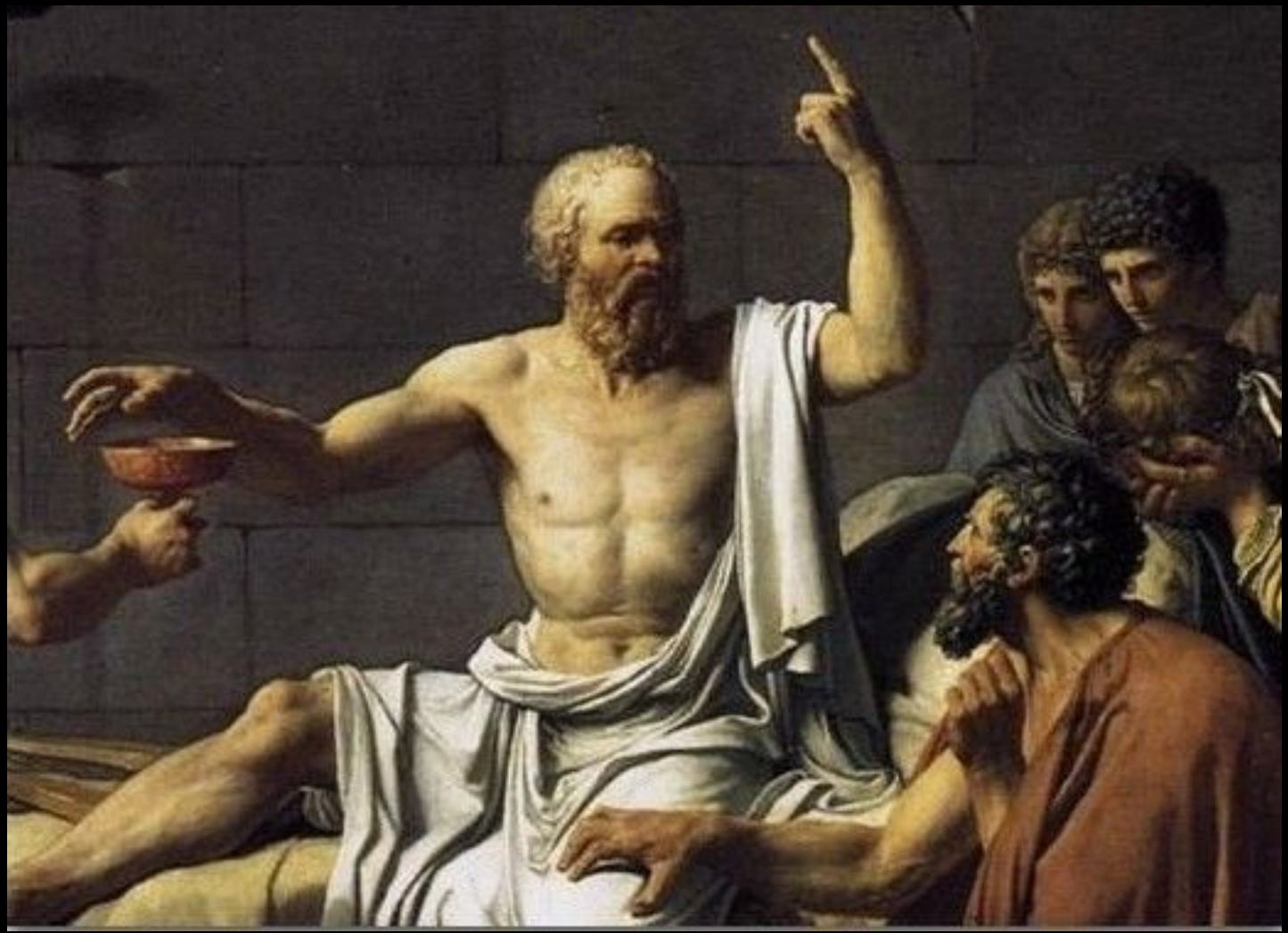
When is benefit(reading) high?

- Many readers
- Long time period
- Lots of data to describe

When is cost(writing) high?

- Many readers -> many versions
- Long time period -> many changes to track
- Lots of data to describe

great_expectations



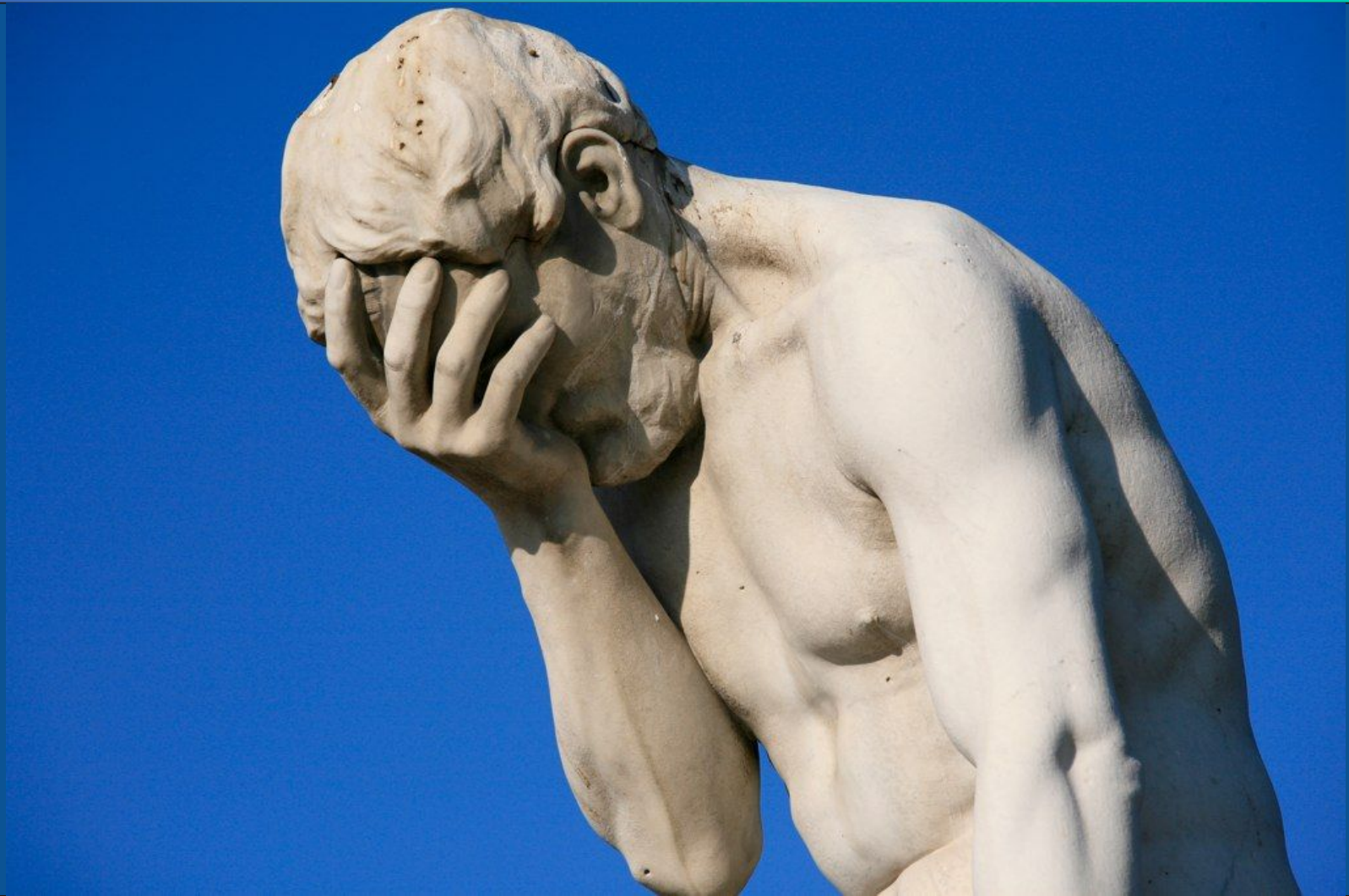
The Law of the Stale Wiki

1. The value of data documentation is a function of stakeholders x data complexity.
2. The cost of manually documenting data is also a function of stakeholders x data complexity.
3. The benefits are shared by many.
4. The cost is borne by a few.

Therefore!

5. As stakeholders and data complexity grow, time spent on data documentation will always grow.
6. ...and always fall further behind the actual need.

great_expectations



What if documentation wasn't manual?

great_expectations

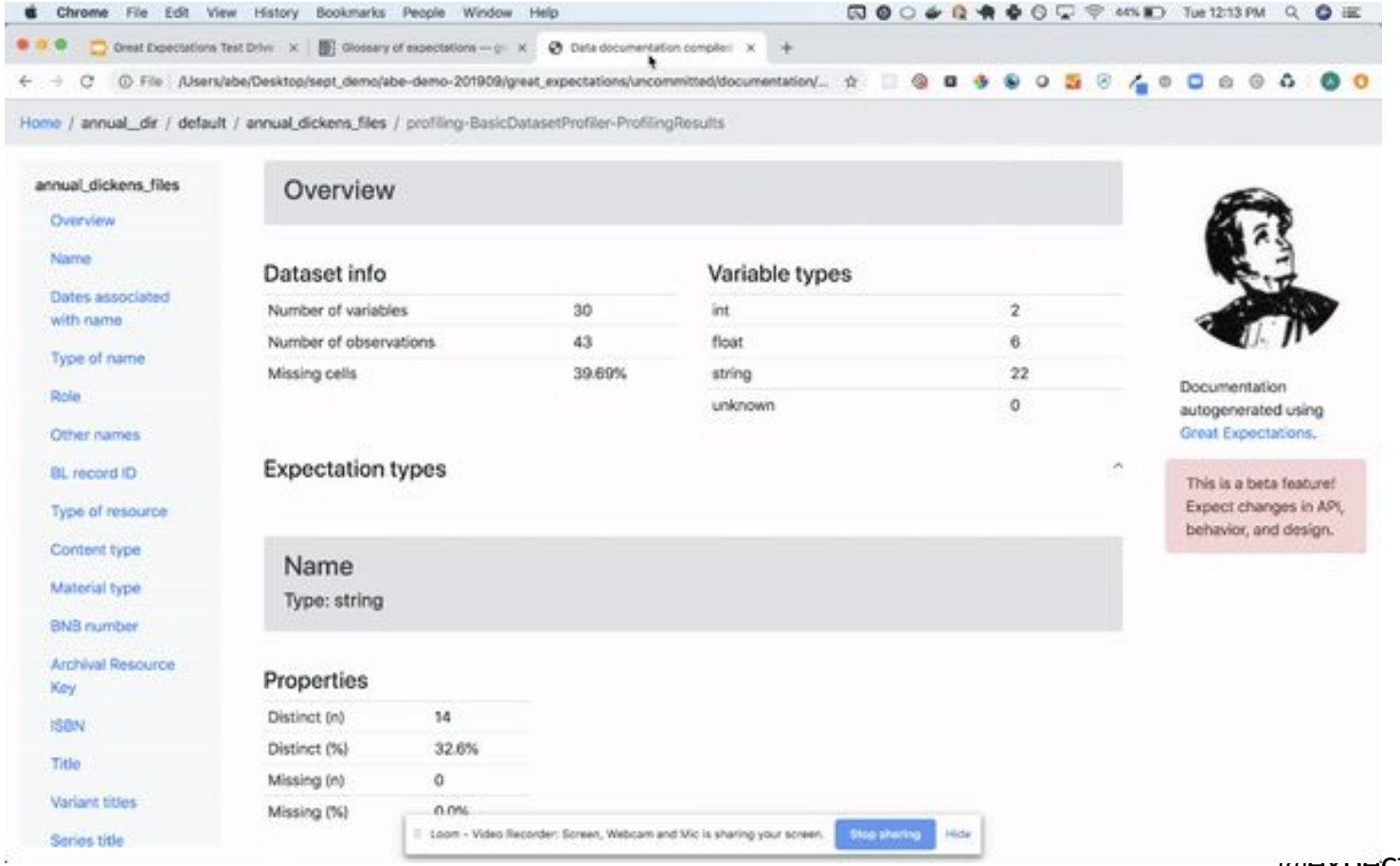
Your docs are your tests, and your tests are your docs.

```
expect_column_values_to_be_between(
    column="room_temp",
    min_value=60,
    max_value=75,
    mostly=.95
)
```

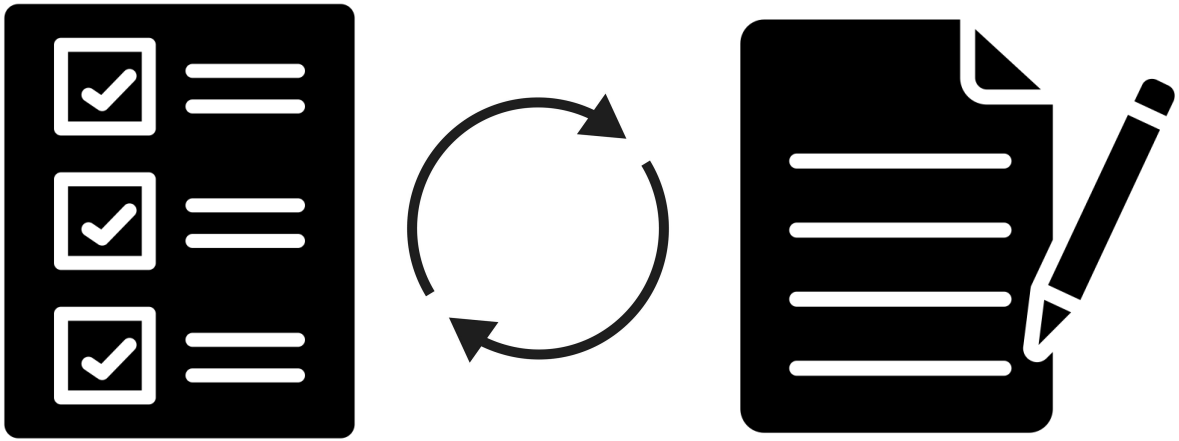

"Values in this column should be between 60 and 75, at least 95% of the time."

"Warning: more than 5% of values fell outside the specified range of 60 to 75."

great_expectations


Chrome File Edit View History Bookmarks People Window Help
Great Expectations Test Driv
Glossary of expectations
Data documentation compiled
File /Users/abe/Desktop/sept_demo/abe-demo-201909/great_expectations/uncommitted/documentation/...
Home / annual__dir / default / annual_dickens_files / profiling-BasicDatasetProfiler-ProfilingResults
annual_dickens_files
Overview
Name
Dates associated with name
Type of name
Role
Other names
BL record ID
Type of resource
Content type
Material type
BNB number
Archival Resource Key
ISBN
Title
Variant titles
Series title
Overview
Dataset info
Number of variables 30
Number of observations 43
Missing cells 39.69%
Variable types
int 2
float 6
string 22
unknown 0
Expectation types
Name
Type: string
Properties
Distinct (n) 14
Distinct (%) 32.6%
Missing (n) 0
Missing (%) 0.0%
Documentation autogenerated using Great Expectations.
This is a beta feature! Expect changes in API, behavior, and design.
Loom - Video Recorder: Screen, Webcam and Mic is sharing your screen.
Stop sharing
Hide

Your docs are your tests,
and your tests are your docs.



great_expectations

Icons created by SBTS from Noun Project

@expectGreatData

Recap

great_expectations

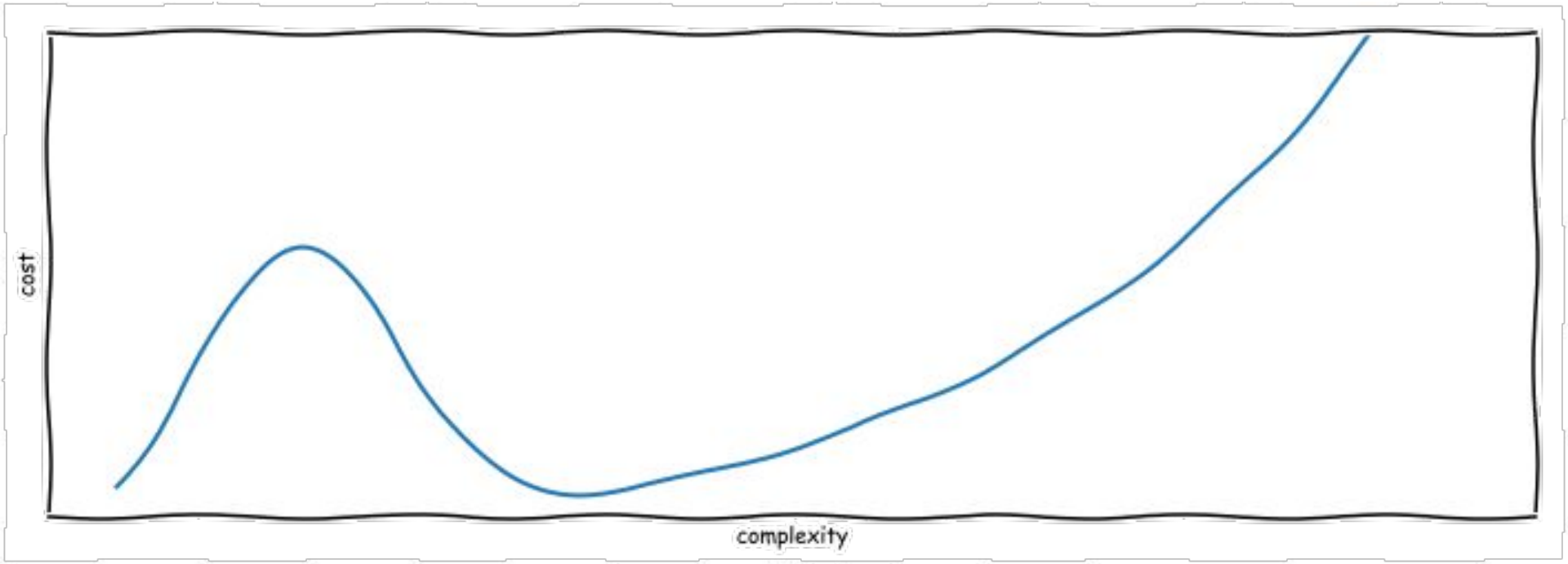
@expectGreatData

What is pipeline debt?

Technical debt in data pipelines, mainly as a result of missing tests and documentation.

great_expectations

Creeping, exponential maintenance burden



great_expectations

@expectGreatData

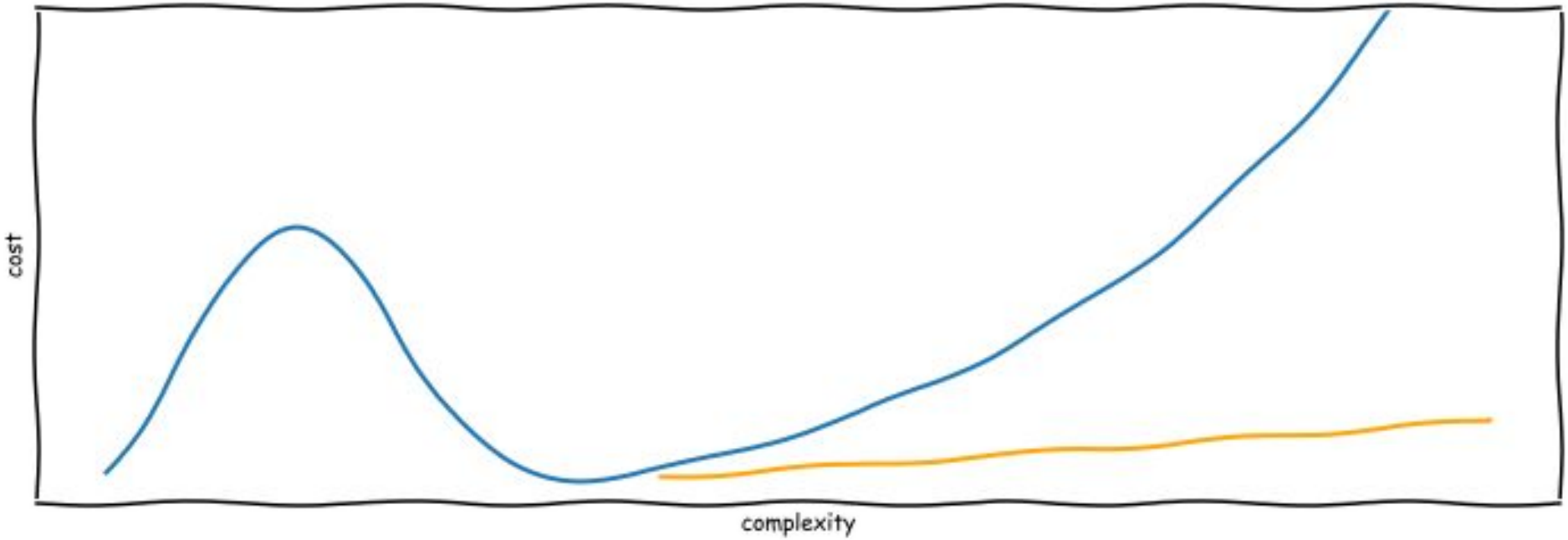
How does Great Expectations beat pipeline debt?

Tests bend the maintenance curve,
making it sublinear instead of superlinear.

great_expectations

@expectGreatData

Tests bend the maintenance curve



great_expectations

@expectGreatData

The Law of the Stale Wiki

As data complexity grows, time spent on data documentation will always grow.

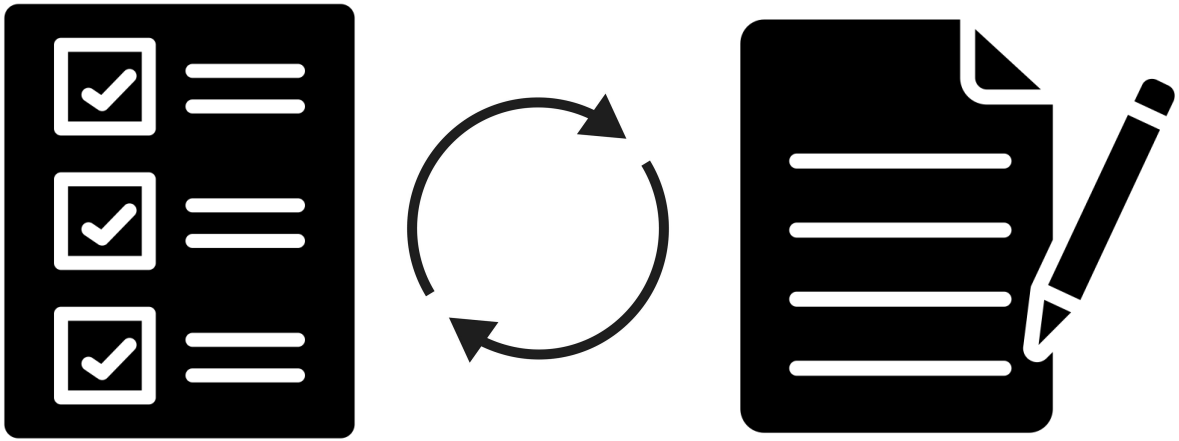
… And always fall further behind.

… Unless you automate your data docs.

great_expectations

@expectGreatData

Your docs are your tests,
and your tests are your docs.



great_expectations

Icons created by SBTS from Noun Project

@expectGreatData

Thank you!

great_expectations

https://greatexpectations.io

How do I get started with Great Expectations?

https://www.greatexpectations.io

1. Say hi and meet others on Slack
   https://greatexpectations.io/slack
2. Review and contribute to code on Github
   https://github.com/great-expectations/great_expectations
3. Ask questions on Discourse
   https://discuss.greatexpectations.io
4. Read the docs
   https://docs.greatexpectations.io
5. pip install great_expectations
   Good python pun? Or best *python pun?*

great_expectations

@expectGreatData