BUILDING ROBUST AND COLLABORATIVE CODED ENGINEERING MODELS

J.D. LANDGREBE, DATA-DELVE LLC

jdlandgrebe@data-delve.com

FORMER PRINCIPAL ENGINEER, PROCTER AND GAMBLE CO. INC.

Colloquium on occasion of Professor Sotiris Pratsinis retirement

Particle Technology Laboratory. Department of Mechanical & Process Engineering, ETH Zürich, Switzerland

February 26, 2025

INTRO

- 30+ years Procter and Gamble technologist/Principal Engineer – Consumer Packaged Goods R&D
- BS ChE (Purdue University USA 1984)
- MS ChE (U. of Cincinnati USA/Pratsinis 1989)

  Thesis: Gas-phase Particulate Manufacture: The Interplay of Chemical Reaction and Aerosol Coagulation
- Data science and technical modeling consulting practice (datadelveengineer.com)
- Adjunct Professor, University of Delaware USA Chemical Engineering

Data Delve

SOME THINGS I LEARNED FROM "DR. P"*

* My then newlywed wife's name for Sotiris circa 1987

- Penetrate technically to the full extent of the topic

230 documents have cited:

A discrete-sectional model for particulate production by gas-phase chemical reaction and aerosol coagulation in the free-molecular regime
Landgrebe J.D., Pratsinis S.E.
(1990) *Journal of Colloid And Interface Science*, 139 (1), pp. 63-86.

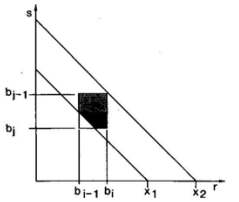
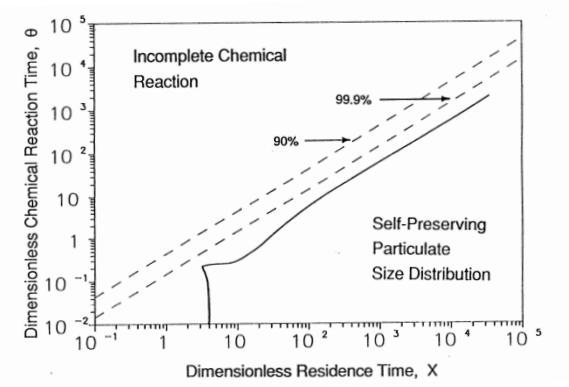


FIG. A1. Collision integral integration limits. The discrete sizes b_{i-1}, b_i, b_{j-1}, and b_j represent the discrete size upper and lower boundaries of two sections, i and j; x_1 and x_2 represent the boundaries of a third section. The shaded area shows combinations of i- and j-section particles resulting in a particle in the third section.

- Research is a team sport. Be collaborative and even friends with your colleagues wherever they may be
- Great visuals rule the day (Corollary: "Say it with dimensionless numbers")



- Mentoring means giving your mentees/those under you a "seat at the table"
  - Conferences
  - Consulting
- Lead a balanced life that pays full attention to our creator and to those around us

With congratulations, gratitude, and warmest best wishes to Sotiris and Eleni!!

Sue and J.D. Landgrebe

WHY BEST PRACTICES FOR CODED MODELS?

- Have you ever:
  - Picked up a colleague's, former student's (or your own 🤪😫) model/code and spent hours trying to understand it?
  - Had doubts or uncertainty about the validity of someone else's model?
  - Had difficulty extending or scaling a previous model (your own or someone else's?)
- HUGE opportunity exists to set best practices individually, in research groups and corporate teams –it's a leadership job!!!

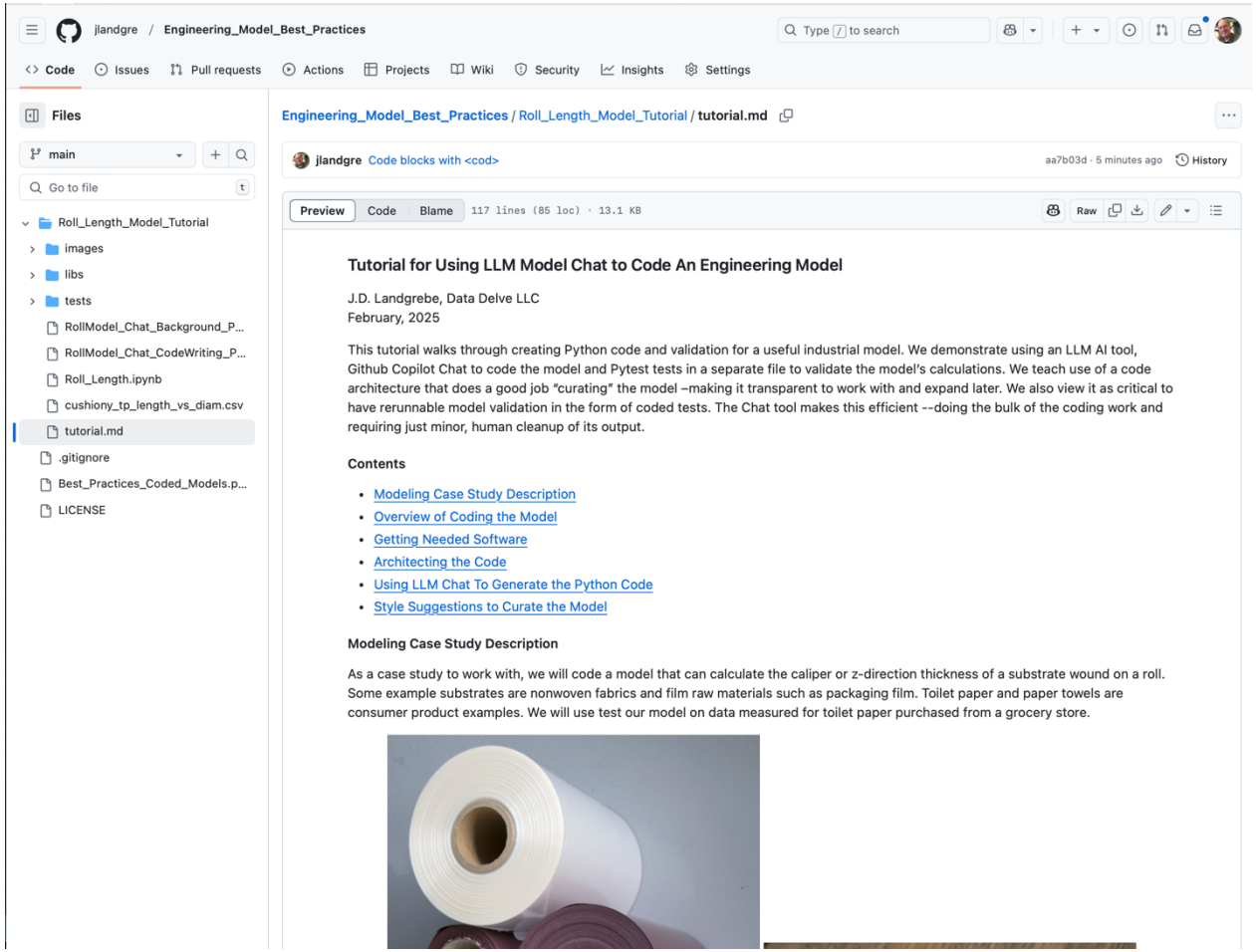
The right amount of code planning cuts down on rework and allows you to work with more complex topics

Picking a model back up from someone else (or from the one-year-ago you) can be challenging

Most models don't have an inspectable validation trail that that builds confidence in them as you work with them and seek to extend them

<u>Target audience:</u>

- Leaders of groups who want set culture for how things are done under their watch
- Individual modelers –graduate students, practicing engineers and scientists. Anyone who builds coded models but is not full-time software engineer or member of software development team



POSTED TUTORIAL

- See full story with tutorial at: https://github.com/jlandgre/Engineering_Model_Best_Practices
- If unfamiliar with Github, click on **Readme.**md doc for download instructions

WHY BEST PRACTICES FOR CODED MODELS?

Best Practices for Models

Culture:
"This is how we do things around here"
aka TIHWDTAH

Objectives of Implementing best practices

1. Gain ability to create Curated coded models personally and in your team

   (see The Fourth Paradigm: Data-Intensive Scientific Discovery, Hey et al., Microsoft Research, 2009)

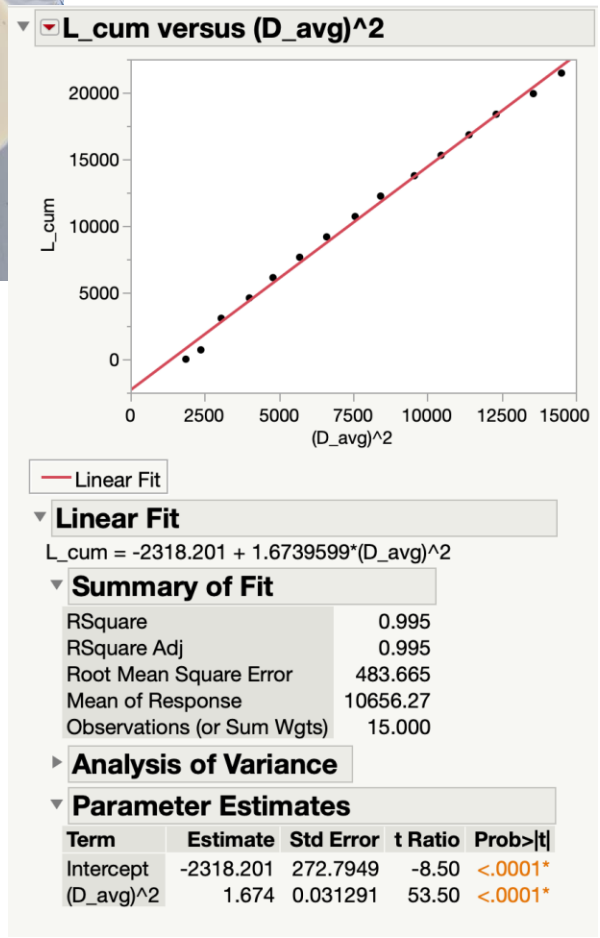
2. Validate with rerunnable tests – guard against future broken models

   (see Test-Driven Design and here inspiration but that's a deep and debated topic. Keep it practical!)

3. Learn to develop efficiently using AI tools to write the code and tests (≈90%)

BEST PRACTICES – PYTHON MODEL

- Github has tutorial to build, curate and validate an industrially useful model for calculating substrate roll properties (Length, L vs. Diam., D)
- A simple, technical example to teach software practices
- Tutorial includes how to use AI to write code for the model and for rerunnable Python (Pytest) tests in separate files



$$L = \frac{\pi(D_{roll}{}^2 - D_{core}{}^2)}{4C} \Rightarrow L = \left(\frac{\pi}{4C}\right) D_{roll}{}^2 + \left(\frac{-\pi D_{core}{}^2}{4C}\right) \Rightarrow Y = mX + b$$

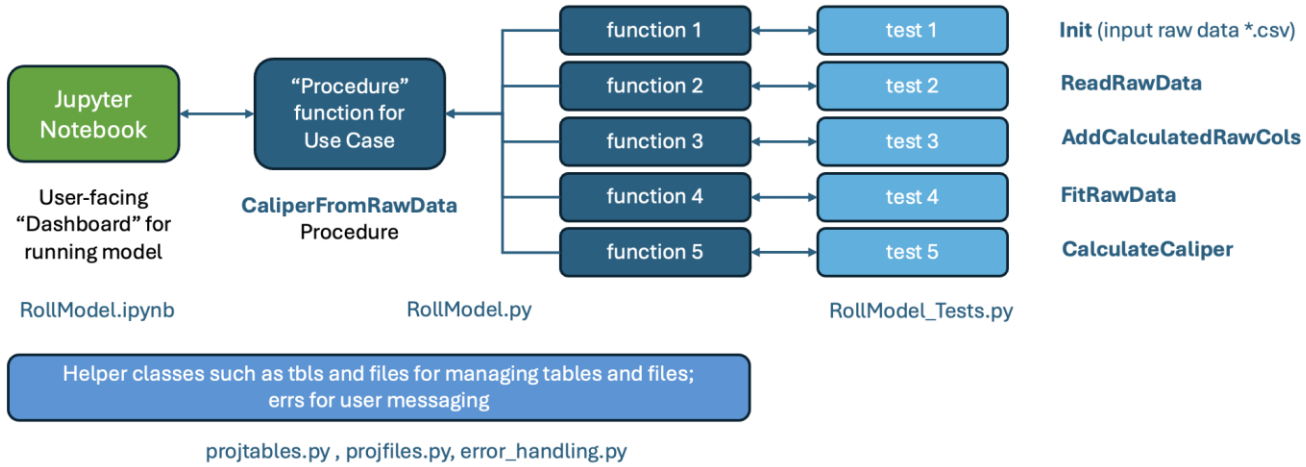
C = material's caliper or thickness

CURATING – PYTHON MODEL EXAMPLE

Curating ≠ long, multi-step code with obscure names

- Code should be <u>easily</u> inspectable by non-coding leader/manager
- Macroscopic, predictable architecture helps curate
- Single-action functions (Clean Code: A Handbook of Agile Software Craftsmanship", R. Martin, 2008)
  - Easy to test
  - Easy for AI tool to write the code and write rerunnable test(s)
- Object-oriented "Class-based"
  - Easy to bring in trusted, helper libraries for working with data etc.
  - Model's instanced code objects are easy to inspect and debug

A Good Architecture for Model Code



Meandering, "spaghetti" code

Multistep "Procedure"

```
def CaliperFromRawDataProcedure(self):
    """
    Procedure to fit a line to transformed raw, length versus diam data
    and thereby enable calculation of an effective caliper for the
    material on a roll of substrate.

    This use case only uses the file_raw Class input --to read in raw
    data
    """
    self.ReadRawData()
    self.AddCalculatedRawCols()
    self.FitRawData()
    self.CalculateCaliper()
```

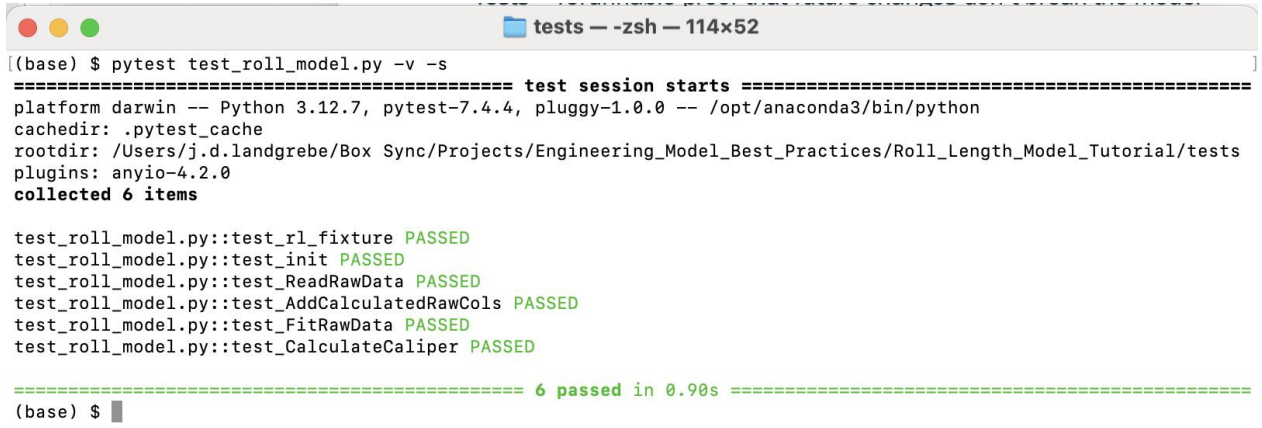
Single-action Function

```
def FitRawData(self):
    """
    Fit line using diam_m^2 as x and length as y
    JDL Feb 18, 2025
    """
    X = self.df_raw[['diam_m^2']]
    y = self.df_raw['length']
    model = LinearRegression().fit(X, y)

    # Set model attributes based on the fit
    self.slope = model.coef_[0]
    self.intcpt = model.intercept_
    self.R_squared = model.score(X, y)
```

TESTING/VALIDATION

Testing is under-utilized by engineers/scientists; yet easy to learn

- Many “validations” consist of feeding a model some inputs, looking at the results and pronouncing it validated
- Tests = rerunnable proof that future changes don’t break the model
- Tests are documentation that explains how each function works
- Standard Python Pytest library manages testing from *.py file separate from model code’s *.py
- Advise minimum one test per model function



Github Copilot Prompt

jlandgre

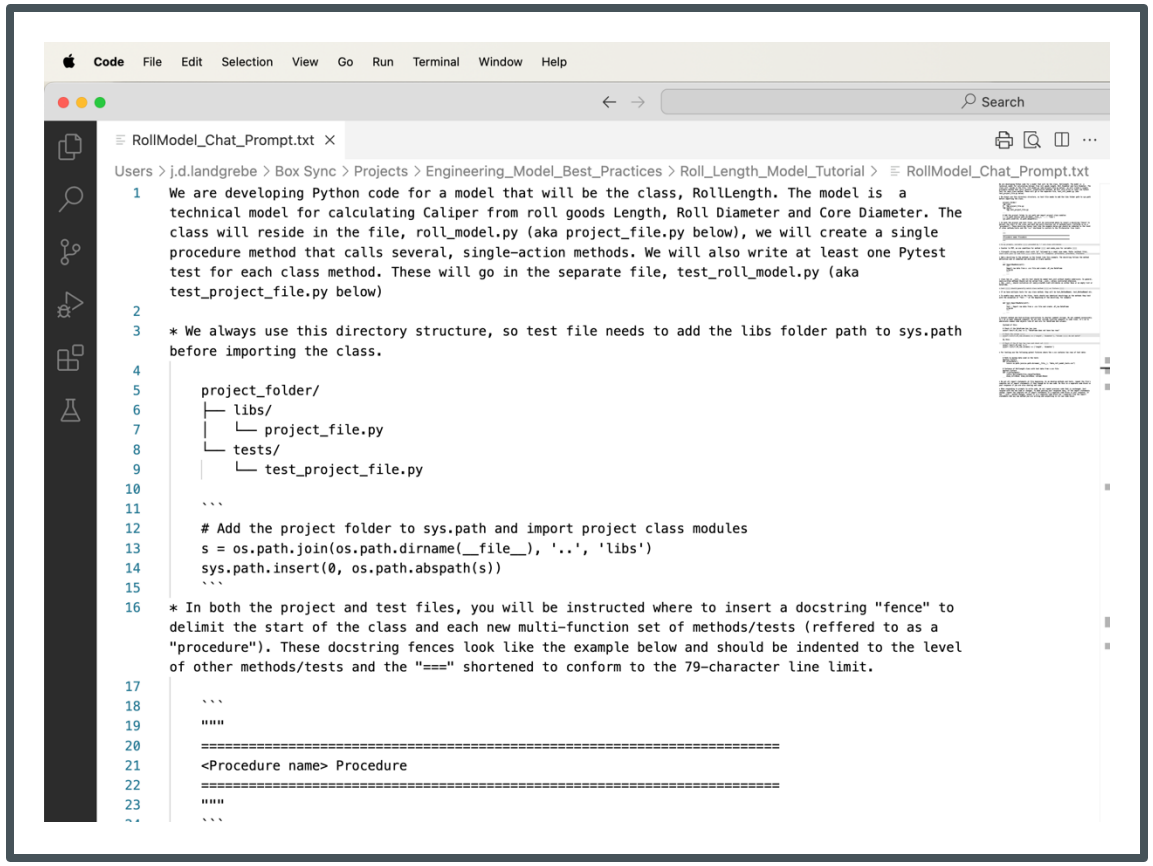
Write test_FitRawData.

- set x and y numpy arrays from the appropriate .df_raw column .values
- calculate delta_y and delta_x from the individual 2-element array values in x and y
- calculate the slope_expected from delta_y and delta_x
- calculate the intercept_expected from y[1], x[1] and slope_expected
- use np.isclose to check that slope_expected and intercept_expected match hand calculated values of 1562.5 and -2.5 to within +/- 0.001
- use np.isclose to check that .slope, .intcpt and .R_squared from the method match slope_expected, intercept_expected and 1.0 respectively

Resulting Pytest Function

```
def test_FitRawData(rl):
    """
    Test - Fit line using diam_m^2 as x and length as y
    JDL 18/2/2025
    """
    rl.ReadRawData()
    rl.AddCalculatedRawCols()
    rl.FitRawData()

    #Numpy arrays for manual calculations
    x, y = rl.df_raw['diam_m^2'].values, rl.df_raw['length'].values

    slope_expected = (y[1] - y[0]) / (x[1] - x[0])
    intercept_expected = y[1] - slope_expected * x[1]

    # Check expected calcs versus hand calculations
    assert np.isclose(slope_expected, 1562.5, atol=0.001)
    assert np.isclose(intercept_expected, -2.5, atol=0.001)

    # Check attributes from FitRawData method
    assert np.isclose(rl.slope, slope_expected, atol=0.001)
    assert np.isclose(rl.intcpt, intercept_expected, atol=0.001)
    assert np.isclose(rl.R_squared, 1.0, atol=0.001)
```

AI TOOLS FOR CODED MODELS



- Organization should provide subscription and not-so-subtly encourage use ("Can you please also post your background prompt for Copilot?)
- Github Copilot Chat + Microsoft VS Code is a good option
- [Typical] Use Case 1 = one-time queries like "How to write a custom, Python iterator for to iterate over a range of DataFrame cells?"
- [More valuable] Use Case 2 = "Write the **FitRawData** function and its Pytest **test_FitRawData** that creates mockup data and checks the linear fit…"
  - **Previous best practices of object-oriented and modular, single-action functions + tests are an ideal fit with using current AI chat tools**
  - Need to supply tool with background to explain "how life works." Cbe standard, pasteable text file (tutorial example)
  - Generally requires more pre-planning than historical coding practices
  - Does not eliminate need for knowing coding –AI tools make mistakes and/or can stray from background intent

CONCLUSIONS

- People join organizations with a range of coding skills and practices
- Leveling up required with seminars, required trainings etc.
- Valuable to set cultural best practices for curation, validation and use of AI tools for coding models
- Heading down this path leads to high efficiency and trustworthiness of coded engineering and scientific models