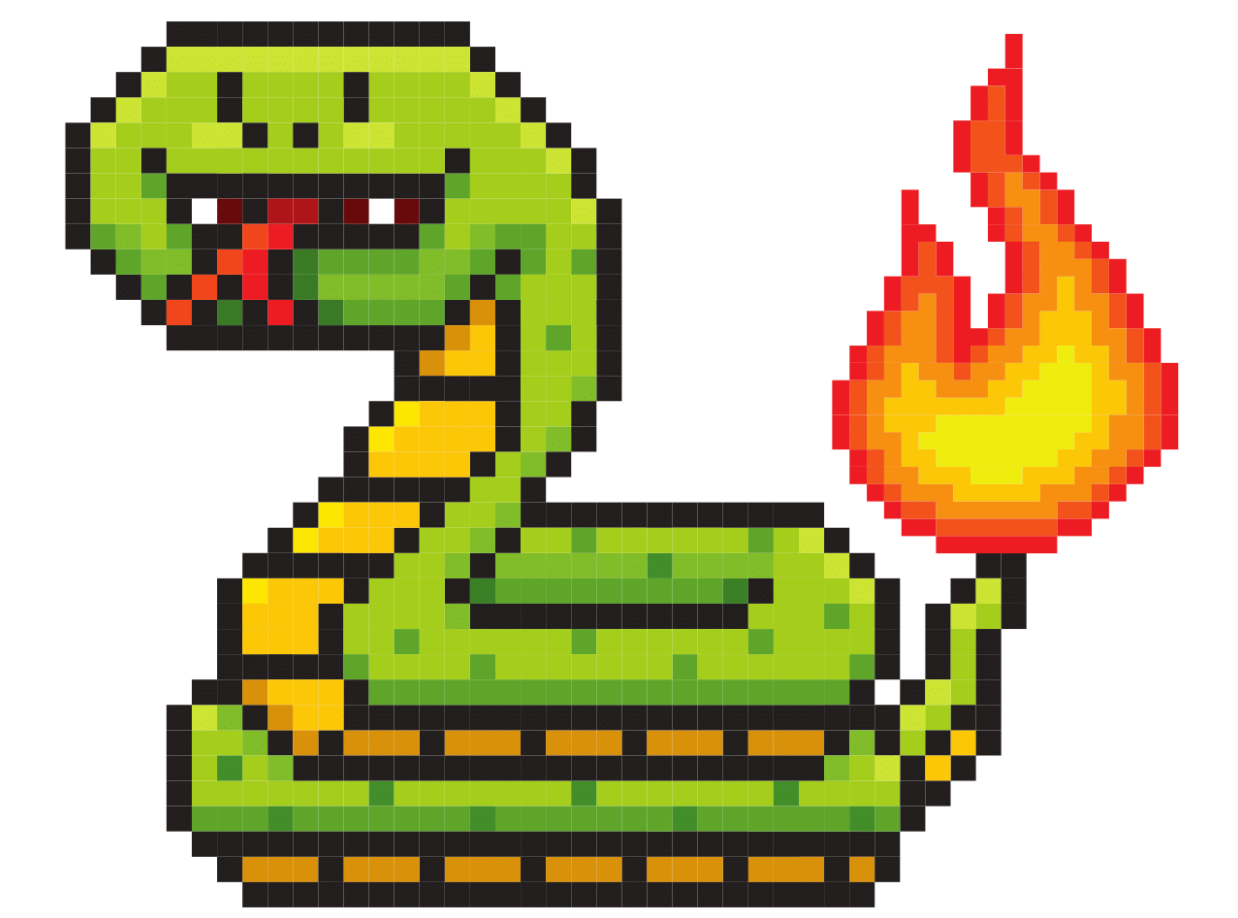




PyKED: a Python-based tool supporting data analysis and experimental reproducibility in combustion



Kyle E. Niemeyer & Bryan W. Weber
Oregon State University University of Connecticut

The problem

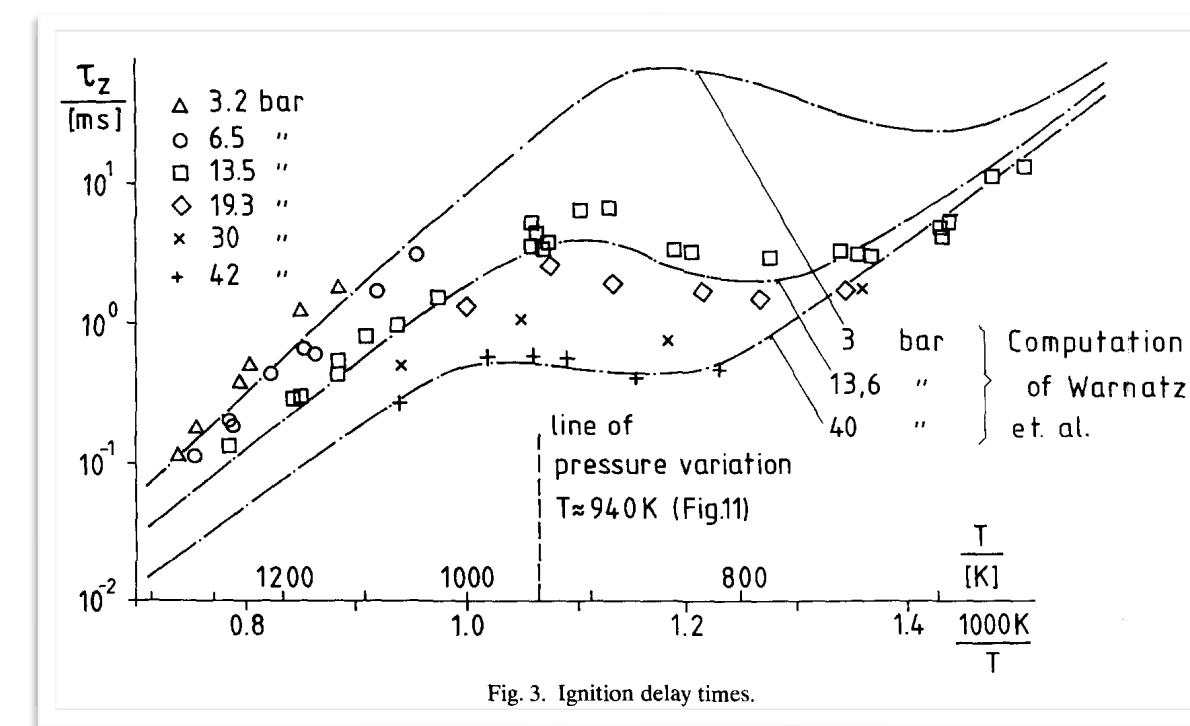
Experiments are good! 😊

There are on the order of millions of data available in the literature 🥰

Sharing is hard 😞

Series	Shock No.	Composition % C ₇ H ₁₆ O ₂	P ₁ torr	P ₂ atm	T ₅ K	P ₂ /P ₁ mm/μsec	τ μsec
A	6	1 11	116	4.66	1260	7.27	.3778
	10	1 11	105	5.17	1410	7.95	.3913
	14	1 11	103	4.52	1323	7.56	.3835
B	22	1 11	50	2.03	1268	7.30	.3785
	29	1 11	70	3.15	1341	7.64	.3851
	37	1 11	50	3.08	1602	6.75	.4077
C	41	0.5 11	100	3.81	1311	6.62	.4014
	45	0.5 11	101	3.51	1245	6.37	.3946
	57	0.5 11	100	4.81	1503	7.30	.4206

PDF table⁵



Figure⁶

```
# n-heptane ignition delay from Colket and Spadaccini 2001
# P (atm), T (K), Ignition Delay (μs)
# Mole Fraction nC7H16 O2 Ar : 0.00192 0.04224 0.95584
7.72 , 1393 , 85
7.78 , 1299 , 345
7.04 , 1235 , 631
6.38 , 1299 , 348
7.53 , 1372 , 134
6.08 , 1236 , 678
7.35 , 1340 , 148
6.63 , 1328 , 211
6.94 , 1395 , 89
```

CSV file⁷

Kyle Niemeyer
To: jeff.borgherson@mcgill.ca
request for data from 2010 alcohol shock tube paper
August 19, 2014 at 7:38 PM
Hello,
I'm currently trying to evaluate the performance of various reaction mechanisms for their ability predict ethanol autoignition, and to that end the ethanol ignition delay data from your 2010 Ene

Email plea

Existing data formats are primarily XML-based and not available under permissive licenses

Existing formats do not have tools to validate and interact with data

Our solution(s)

Permissive licensing and community-based management of the database and schema

ChemKED: Chemical Kinetics Experimental Data format
+

PyKED: Python software for working with Chem**KED** files

ChemKED

Written in YAML: human- AND machine- readable AND writable!

Parsers and libraries for nearly every programming language!

Uncertainty

ChemKED supports specification of uncertainties for **all** quantities in a data point

Dimensionless: Composition (mole/mass fraction, mole percent)

Dimensional: temperature, pressure, ignition delay, etc.

Supports relative and absolute uncertainty

PyKED

Python-based tool to validate ChemKED files

Enables interaction with experimental data

```
file-authors:-
  - name: Kyle E Niemeyer
    ORCID: 0000-0003-4425-7097
file-version: 0
chemked-version: 0.4.1
reference:-
  doi: 10.1016/j.ijhydene.2007.04.008
authors:-
  - name: N. Chaumeix
  - name: S. Pichon
  - name: F. Lafosse
  - name: C.-E. Paillard
journal: International Journal of Hydrogen Energy
year: 2007
volume: 32
pages: 2216-2226
detail: Fig. 12., right, open diamond
experiment-type: ignition delay
apparatus:-
  kind: shock tube
  institution: CNRS-ICARE
  facility: stainless steel shock tube
common-properties:-
  pressure: &pres-
    - 220 kilopascal
  composition: &comp-
    kind: mole fraction
  species:-
    - species-name: H2
      InChI: 1S/H2/h1H
      amount:-
        - 0.00444
    - species-name: O2
      InChI: 1S/O2/c1-2
      amount:-
        - 0.00556
    - species-name: Ar
      InChI: 1S/Ar
      amount:-
        - 0.99
  ignition-type: &ign-
    target: pressure
    type: d/dt max
datapoints:-
  - temperature:-
    - 1164.48 kelvin
    - uncertainty-type: absolute
    - uncertainty: 10 kelvin
  - ignition-delay:-
    - 471.54 us
  - pressure:-
    - 220 kilopascal
  composition:-
    kind: mole percent
  species:-
    - species-name: H2
      InChI: 1S/H2/h1H
      amount:-
        - 0.444
    - uncertainty-type: relative
    - uncertainty: 0.01
```

```
DOI: 10.5281/zenodo.1194920
license: BSD
codecov: 100%
code of conduct: contributor covenant
build: passing
build: passing
Anaconda Cloud: 0.4.1
```

Example: Working with Data

PyKED enables **interaction** with data

```
In [1]: from pyked import ChemKED
        from pyked.utils import units
        import pandas as pd
        import matplotlib.pyplot as plt
        from pathlib import Path
        import numpy as np
        from sklearn import linear_model
        %matplotlib notebook

In [2]: buoh = pd.concat(
        [ChemKED(name).get_dataframe() for name in Path('/Users/bryan/GitHub/ChemKED-database').glob('**/*.yaml')],
        ignore_index=True
        )

In [3]: buoh['Temperature'] = buoh['Temperature'].apply(lambda x: x.to('kelvin').magnitude)
        buoh['Ignition Delay'] = buoh['Ignition Delay'].apply(lambda x: x.to('seconds').magnitude)
        buoh['Pressure'] = buoh['Pressure'].apply(lambda x: x.to('Pa').magnitude)
        buoh['1000/T'] = 1000.0/buoh['Temperature']
```

Convert everything to consistent units automatically!

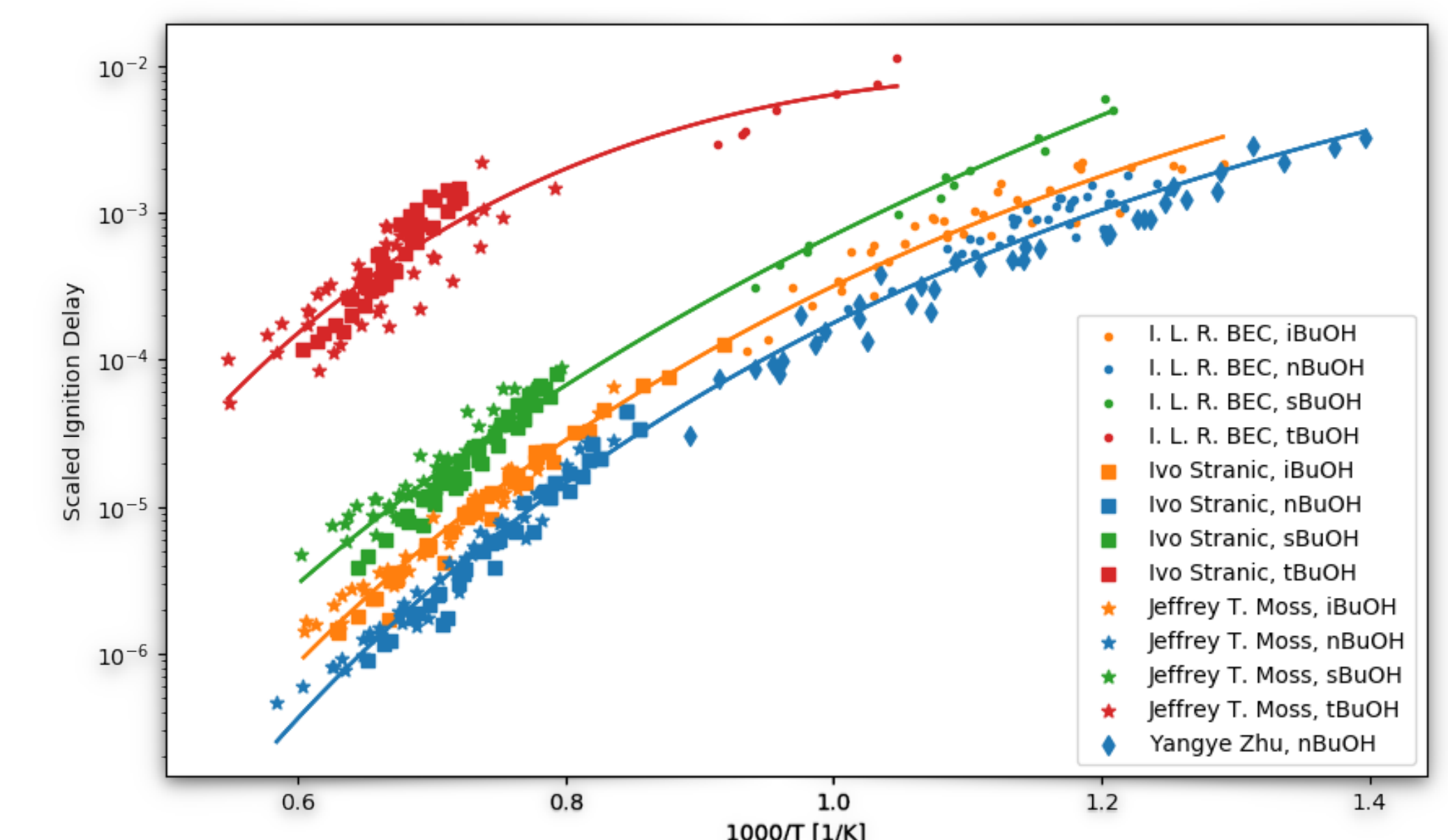
Multiple linear regression

$$\ln \tau = \ln A + n \ln T - \frac{E_a}{RT} + a \ln X_{\text{Fuel}} + b \ln X_{\text{O}_2} + c \ln P$$

```
In [5]: models = {}
        for name, group in buoh.groupby('Fuel'):
            Y = np.log(group['Ignition Delay'])
            X = pd.DataFrame(columns=['log_T', '1/RT', 'log_Fuel', 'log_O2', 'log_P'], dtype=np.float64)
            X['log_T'] = np.log(group['Temperature'])
            X['log_P'] = np.log(group['Pressure'])
            R = (1.0*units.molar_gas_constant).to('J/(mol*K)').magnitude
            X['1/RT'] = -1/(R*group['Temperature'])
            X['log_Fuel'] = np.log(group['Fuel_X'])
            X['log_O2'] = np.log(group['O2']).apply(lambda x: x.magnitude)
            model = linear_model.LinearRegression()
            model.fit(X.as_matrix(), Y.as_matrix())
            n, Ea, a, b, c = model.coef
            A = np.exp(model.intercept_)

            invT_max = group['1000/T'].max()
            invT_min = group['1000/T'].min()
            invT = np.linspace(invT_min, invT_max, 100)
            T = 1000/invT

            models[name] = {'A': A, 'n': n, 'Ea': Ea, 'a': a, 'b': b, 'c': c, 'T': T, 'invT': invT}
```



Check out
the docs



<https://github.com/pr-omethe-us/PyKED>