Teaching Data Science to Undergraduate Students in the University of Iowa

Jun Wang

Jun-wang-1@uiowa.edu

http://arroma.uiowa.edu

With contributions from:

Charles Stanier, Joe Gomes

and many colleagues

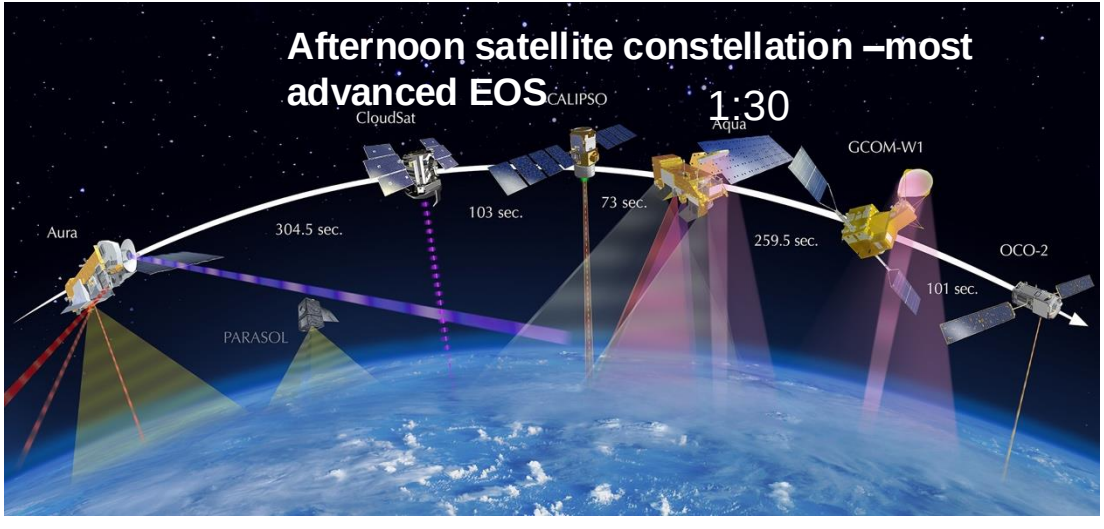
Department of Chemical & Biochemical Engineering
The University of Iowa

Teaching Data Science to Students and Teachers I
Wednesday, October 30, 2024
AIChE 2024

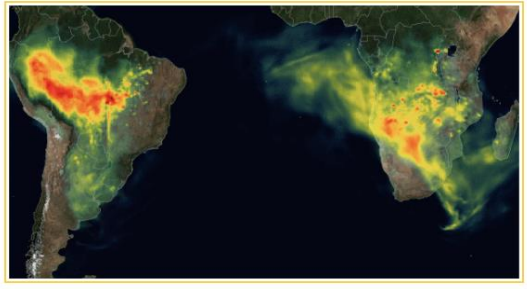
A bit intro about me

Research

- Satellite remote sensing of atmospheric composition and fires
- Smart & connected sensors for canopy parameters via IoT
- Global and regional modeling/prediction of atmospheric composition, air-land interaction, and climate change.
- Data assimilation & big data research.
- Dedicated 1 Peta-byte data storage + 1200 CPUs
- A few GPU nodes as well.

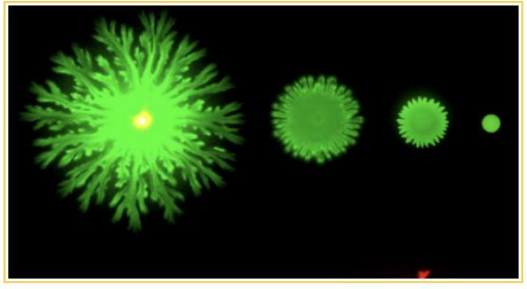


CBE Departmental Research Focus Areas



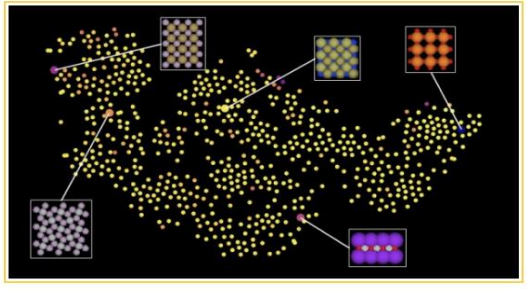
Air Quality & Climate

Learn More



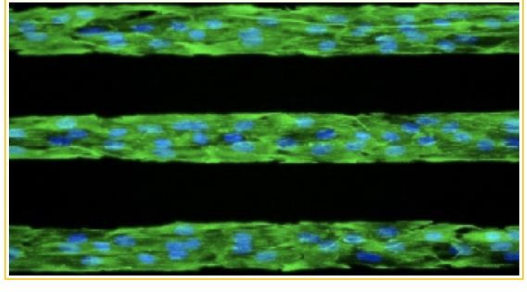
Biological and Pharmaceutical

Learn More



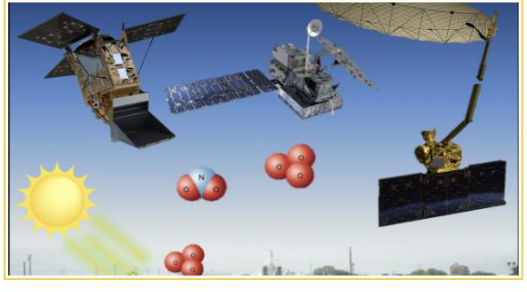
Machine Learning & Simulation

Learn More



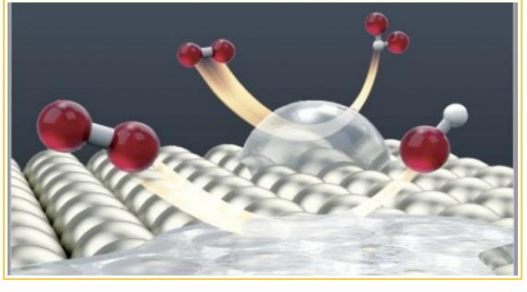
Polymer & Advanced Material

Learn More



Remote & Smart Sensing

Learn More



Sustainable Energy & Clean Water

Learn More

1st Year	Session	Course	Course Name	SH	P: Prerequisite; C: Coreq; R: Recommend
	ALL	MATH:1550	Engineering Math I: Single Variable Calculus	4	P: MPT Level 3 score of 9 or higher or ALEKS score of 75 or higher
	F	ENGR:1100	Introduction to Engineering Problem Solving	3	
	ALL	CHEM:1110	Principles of Chemistry I	4	
	ALL	RHET:1030	Rhetoric	4	
	F	ENGR:1000	Engr Success for First Year Students	1	First Semester Standing
			Total	16	
	ALL	MATH:1560	Engineering Math II: Multi-Variable Calculus	4	P: MATH 1550
	F/S	ENGR:1300	Introduction to Engineering Computing	3	C: MATH:1550
	ALL	PHYS:1611	Introductory Physics I	4	C: MATH:1550
	ALL	MATH:2550	Engineering Math III: Matrix Algebra	2	P: MATH:1550
	ALL	CHEM:1120	Principles of Chemistry II	4	P:CHEM:1100
	S	CBE:1000	CBE Departmental Seminar	1	
2nd Year			TOTAL	18	
	ALL	MATH:2560	Engineering Math IV: Differential Equations	3	P:MATH:1560,MATH:2550
	ALL		General Education Component #1	3	
	ALL	ENGR:2110	Engineering Fundamentals I:Statics	2	P:MATH:1550; C:MATH:1560,PHYS:1611
	ALL	ENGR:2120	Engineering Fundamentals II: Circuits	3	C:MATH:2560
	ALL	ENGR:2130	Engineering Fundamentals III: Thermodynamics	3	P:CHEM:1110,PHYS:1611; C:MATH:1560
	F/S	CBE:2105	Process Calculations	3	P:MATH:1550
			Total	17	
	S	CBE:3105	ChE Thermodynamics	3	P:ENGR:2130; C:CBE:2105
	S	CBE:3109	Fluid Flow	2	C:CBE:2105
	ALL	CHEM:2210	Organic Chemistry I	3	P:CHEM:1120
	ALL		General Education Component course	3	
	ALL	ENGR:2720	Material Science	3	P:CHEM:1110; C:MATH:1550
	ALL	STAT:2020	Statistics elective	3	P:MATH:1560
	F/S	CBE:3000	CBE Professional Seminar	1	P:CBE:2105
3rd Year			Total	18	
	F	CBE:3113	Heat & Mass Transfer	3	P:MATH:2560,CBE:2105; R:CBE:3109
	F	CBE:3125	Chemical Process Safety	3	P:CBE:3105,CBE:3109; C:CBE:3113
	F	CBE:3117	Separations	3	P:CBE:2105,CBE:3105; C:CBE:3113
	ALL	CHEM:2220	Organic Chemistry II	3	P:CHEM:2210
	ALL	CHEM:2410	Organic Chemistry Laboratory	3	P:CHEM:2210; C:CHEM:2220
	F/S	CBE:3000	CBE Professional Seminar	1	P:CBE:2105
			Total	16	
	F/S	CBE:3120	Chemical Reaction Engineering	3	P:MATH:2560; C:CBE:3105; R:CBE:3113
	F	CBE:3150	Thermodynamics/Transport Laboratory	3	P:CBE:3105,CBE:3113
	F/S		Elective focus area course	3	
	S	CBE:3205	Introduction to Biochemical Engineering	3	P:CBE:2105; C:CBE:3109; R:CBE:3120
	ALL		General Education Component course	3	
	F/S	CBE:3000	CBE Professional Seminar	1	P:CBE:2105
4Th Year			Total	16	
	F	CBE:4105	Process Dynamics & Control	3	P:MATH:2560,CBE:2105,CBE:3109; C:CBE:3120
	F	CBE:4109	Chemical Engineering Process Design I	2	P:CBE:3109,CBE:3113,CBE:3117; C:CBE:3120,CBE:3125
	S	CBE:3155	Chemical Reaction Engineering/Separation Lab	3	P:CBE:3117; C:CBE:3120; R:STAT:2020
	ALL		Advanced chemistry elective	3	
	F/S		Elective focus area course	3	
	F/S		Elective focus area course	3	
	F/S	CBE:3000	CBE :Professional Seminar	1	P:CBE:2105
			Total	18	
	S	CBE:4110	Chemical Engineering Process Design II	3	P:CBE:4109; R:CBE:4105,CBE:5205
	ALL		Advanced science elective	3	
	F/S		Elective focus area course	3	
	ALL		General Education Component course	3	
	ALL		General Education Component course	3	
	S	CBE:4195	Senior Enriching Activities Seminar	0	C:CBE:4110
			Total	15	

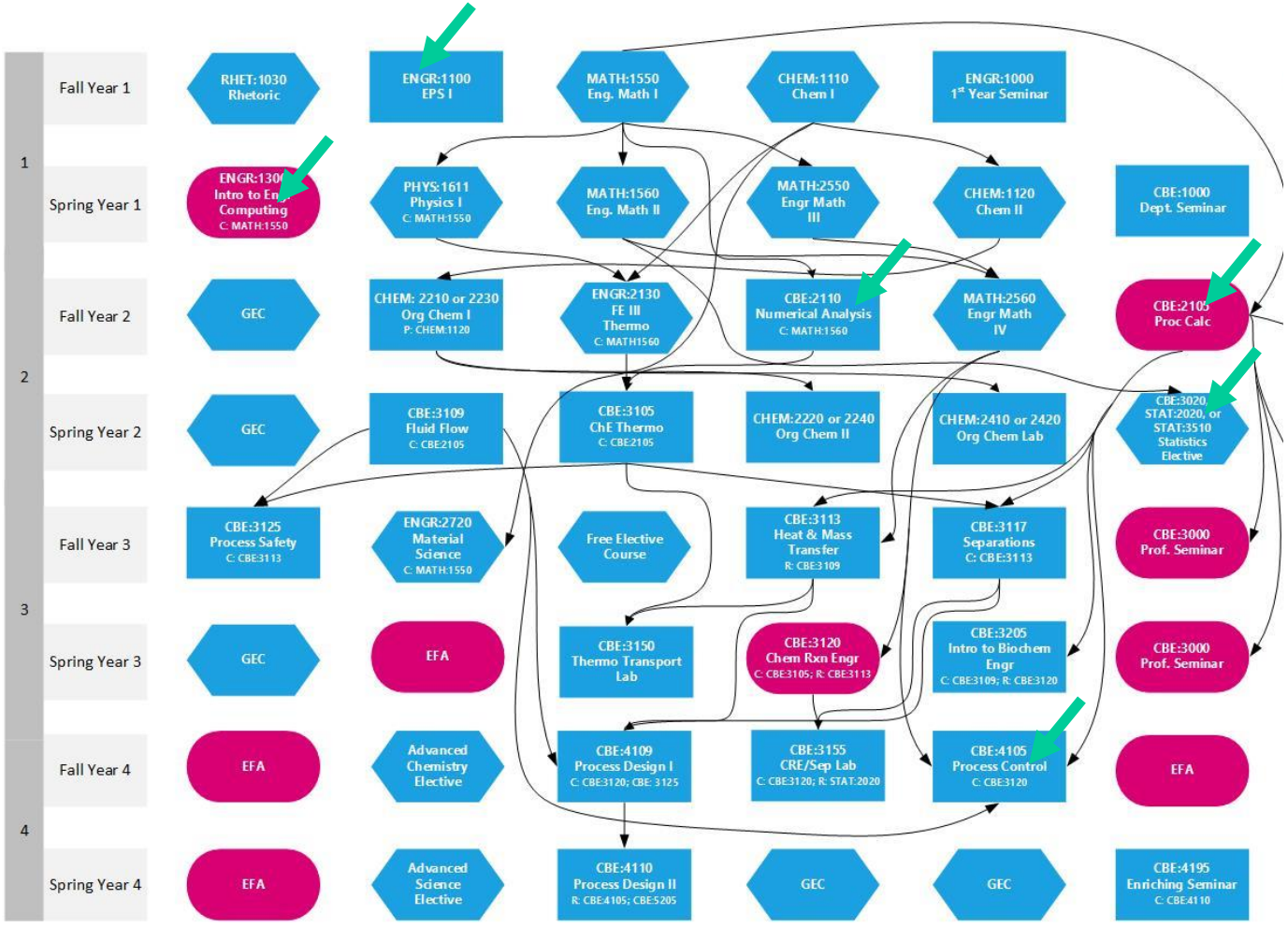
CBE Undergraduate Curriculum

prior to 2019

- Students are taught some skills of programming & computing in first year
  - “Intro. To Engr. Prob. Solving”
  - “Intro. To Engr. Computing”
- Students will use some computing in the second year (“process calculations”)
- After that, students would barely need any data science skills in senior project design.

CBE Undergraduate Curriculum in Univ. of Iowa

Since 2019-2020



Solid foundation

CBE added two into (sophomore year)

- Numerical Analysis
- Statistics

Engr added one Machine learning Intro (freshman year)

Numerical Analysis for Chemical Engineers

CBE 2110, slide materials contributed by Joe Gomes

- First offering Fall 2021 (sophomore-level)
- Created to unify teaching of numerical techniques in Excel and Python that were previously spread across the curriculum.
- Content:
  - Arrays, equations, functions, data, graphs in Python and Excel
  - Interpolation methods, trendlines, curve-fitting, and parameter estimation
  - Solving linear and non-linear (systems of) equation(s)
  - Numerical integration and ordinary differential equations
  - Optimization
- Learning Goals: Be able to
  - Use a spreadsheet package to perform engineering calculations including processing and analysis of data, graphical analysis, and presentation.
  - Demonstrate an understanding of fundamental mathematics (including calculus, linear and nonlinear simultaneous equation solving, etc.) by using computer tools to solve problems.
  - Convert problem solving strategies to procedural algorithms, write program structures, and understand when programming is most appropriate.
  - **Through the years, since its inception, more students like the class to focus more on Python**.

Example (1) (Excel)

Solving non-linear equation by root-finding

Find roots (solution) of this equation

$$x^3 - 10(x-1)^2 = -1$$

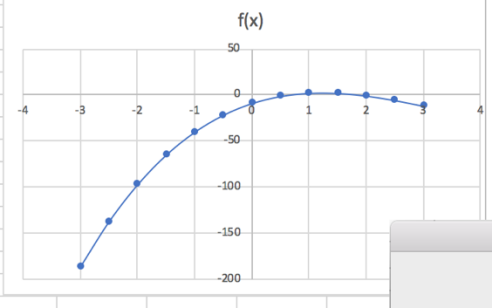
Rearrange to the form f(x) = 0

$$x^3 - 10(x-1)^2 + 1 = 0$$

Plot the function

x	f(x)
0.6440895	0.0004787

x	f(x)
-3	-186
-2.5	-137.125
-2	-97
-1.5	-64.875
-1	-40
-0.5	-21.625
0	-9
0.5	-1.375
1	2
1.5	1.875
2	-1
2.5	-5.875
3	-12

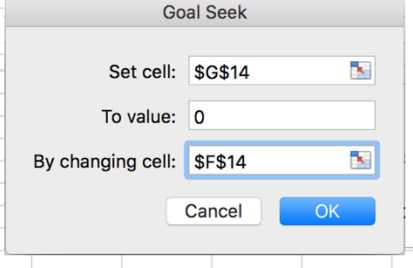


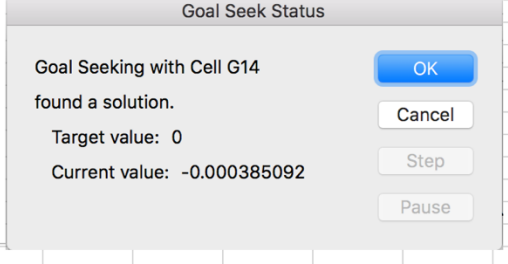
Goal Seek steps:

Step 0 (optional): Generate an initial guess
- Guess and check
- Evaluating/plotting a range of values

Step 1: Define cells containing the dependent and independent variables:
- Dependent variable f(x) (G14) - this cell should be a formula using "x"
- Independent variable x (F14)

Step 2: Run Goal Seek
- Data > What-If Analysis > Goal Seek
- "Set cell" = f(x)
- "To value" = 0
- "By changing cell" = x
- Click "OK"





Example (2) (Python)
Parameter estimation by curve fitting

Exercise: Vapor Pressure of Mercury

Two variables, vapor pressure P and temperature T, are related by the Antoine equation

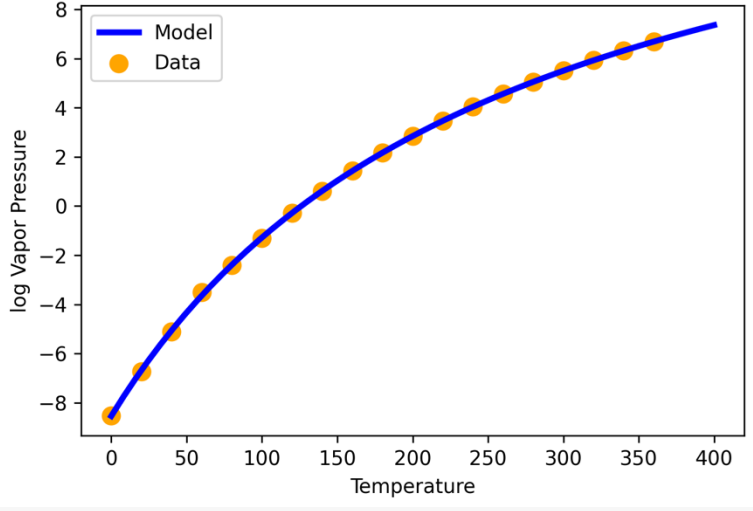
$$\log P = A - \frac{B}{T + C}$$

Given the data set for P and T below, calculate A, B, and C using the method of least squares with `scipy.optimize.minimize`.

```
[9]  vp_df = pd.read_csv('pressure.csv')
     p = vp_df['pressure'].values
     t = vp_df['temperature'].values
     logp = np.log(p)
```

```
[10] def f(params, t, logp):
       A, B, C = params
       d = logp - (A - B/(t+C))
       phi = np.sum(d**2)
       return phi
```

```
[11] sol = minimize(f, x0=[20, 1000, 800], args=(t, logp))
     A, B, C = sol.x
     t_line = np.linspace(0, 400)
     logp_line = A - B/(t_line+C)

     plt.plot(t_line, logp_line, lw=3, c='blue', label='Model')
     plt.scatter(t, logp, s=75, c='orange', label='Data')
     plt.xlabel("Temperature")
     plt.ylabel("log Vapor Pressure")
     plt.legend()
```



Statistics for Chemical & Natural Resource Engr.

CBE 3020 in U. Iowa

Content

- Probability
- Empirical distributions
- Exploratory data analysis
- Parametric probability distributions
- Hypothesis testing
- Statistical forecasting
- Statistical verification
- Time series analysis
- Multivariate statistics
- Advanced techniques (PCA, etc)

Students are asked to

- Write Python code to solve HW
- Write Python code to solve exams

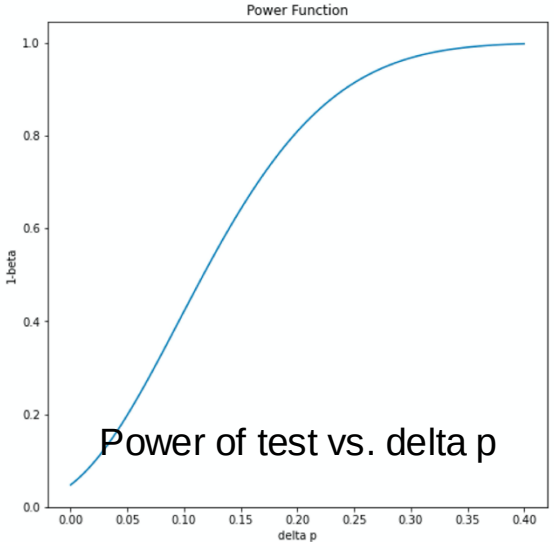
  (allow online access to Python resources & write anything on ¼ size of a letter)
- Use Jupiter Notebook; save solutions in pdf
- Reproduce figures in the textbook with Python
- Write own routines
- Be familiar with the terminology in Python vs. in the textbook

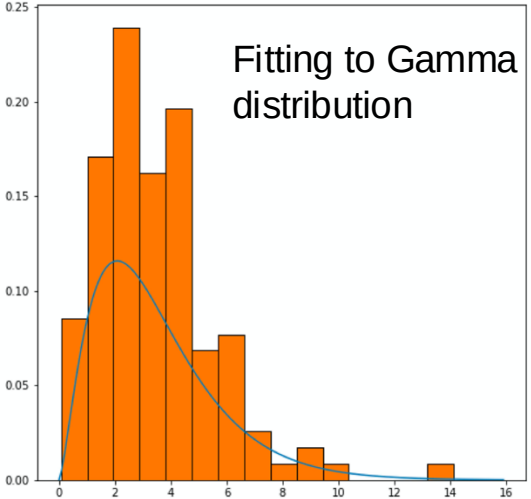
Examples (1)

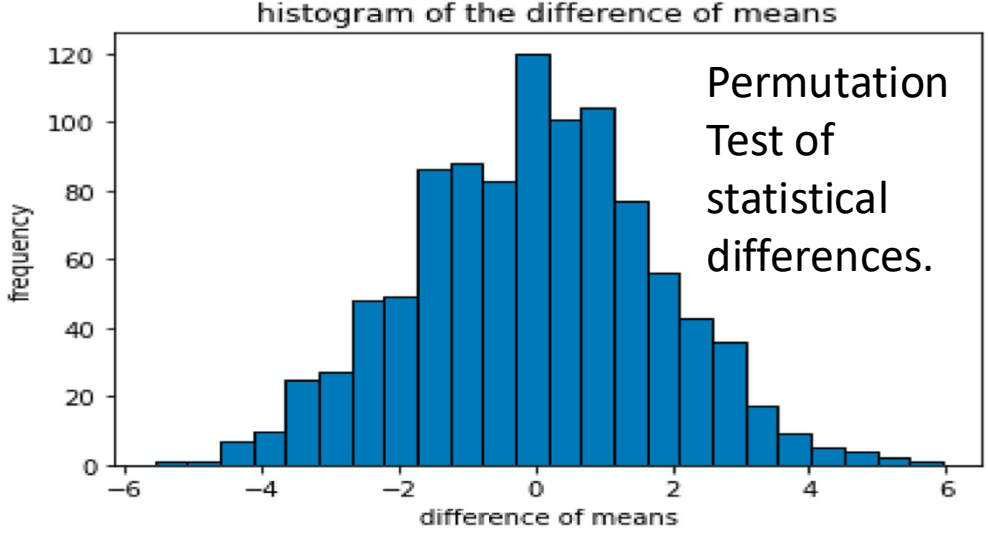
Regenerate left-tail Gaussian Table

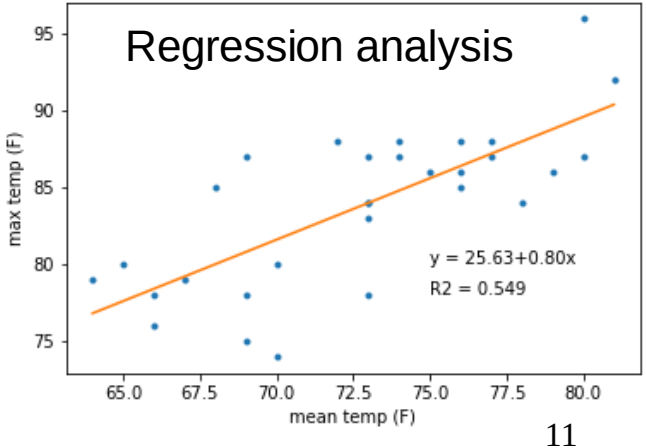
Z	0.09	0.08	0.07	0.06	0.05	0.04	0.03	0.02	0.01	0.00	Z
-4.0	0.00002	0.00002	0.00002	0.00002	0.00003	0.00003	0.00003	0.00003	0.00003	0.00003	-4.0
-3.9	0.00003	0.00003	0.00004	0.00004	0.00004	0.00004	0.00004	0.00004	0.00005	0.00005	-3.9
-3.8	0.00005	0.00005	0.00005	0.00006	0.00006	0.00006	0.00006	0.00007	0.00007	0.00007	-3.8
-3.7	0.00008	0.00008	0.00008	0.00008	0.00009	0.00009	0.00010	0.00010	0.00010	0.00011	-3.7
-3.6	0.00011	0.00012	0.00012	0.00013	0.00013	0.00014	0.00014	0.00015	0.00015	0.00016	-3.6
-3.5	0.00017	0.00017	0.00018	0.00019	0.00019	0.00020	0.00021	0.00022	0.00022	0.00023	-3.5
-3.4	0.00024	0.00025	0.00026	0.00027	0.00028	0.00029	0.00030	0.00031	0.00032	0.00034	-3.4
-3.3	0.00035	0.00036	0.00038	0.00039	0.00040	0.00042	0.00043	0.00045	0.00047	0.00048	-3.3
-3.2	0.00050	0.00052	0.00054	0.00056	0.00058	0.00060	0.00062	0.00064	0.00066	0.00069	-3.2
-3.1	0.00071	0.00074	0.00076	0.00079	0.00082	0.00084	0.00087	0.00090	0.00094	0.00097	-3.1
-3.0	0.00100	0.00103	0.00107	0.00111	0.00114	0.00118	0.00122	0.00126	0.00131	0.00135	-3.0
-2.9	0.00139	0.00144	0.00149	0.00154	0.00159	0.00164	0.00169	0.00175	0.00181	0.00187	-2.9
-2.8	0.00193	0.00199	0.00205	0.00212	0.00219	0.00226	0.00233	0.00240	0.00248	0.00256	-2.8
-2.7	0.00264	0.00272	0.00280	0.00289	0.00298	0.00307	0.00317	0.00326	0.00336	0.00347	-2.7
-2.6	0.00357	0.00368	0.00379	0.00391	0.00402	0.00415	0.00427	0.00440	0.00453	0.00466	-2.6
-2.5	0.00480	0.00494	0.00509	0.00523	0.00539	0.00554	0.00570	0.00587	0.00604	0.00621	-2.5
-2.4	0.00639	0.00657	0.00676	0.00695	0.00714	0.00734	0.00755	0.00776	0.00798	0.00820	-2.4
-2.3	0.00842	0.00866	0.00889	0.00914	0.00939	0.00964	0.00990	0.01017	0.01044	0.01072	-2.3
-2.2	0.01101	0.01130	0.01160	0.01191	0.01222	0.01255	0.01287	0.01321	0.01355	0.01390	-2.2
-2.1	0.01426	0.01463	0.01500	0.01539	0.01578	0.01618	0.01659	0.01700	0.01743	0.01787	-2.1
-2.0	0.01831	0.01876	0.01923	0.01970	0.02018	0.02068	0.02118	0.02169	0.02222	0.02275	-2.0
-1.9	0.02330	0.02385	0.02442	0.02500	0.02559	0.02619	0.02680	0.02743	0.02807	0.02872	-1.9
-1.8	0.02938	0.03005	0.03074	0.03144	0.03216	0.03289	0.03363	0.03438	0.03515	0.03593	-1.8
-1.7	0.03673	0.03754	0.03836	0.03920	0.04006	0.04093	0.04182	0.04272	0.04363	0.04457	-1.7
-1.6	0.04552	0.04648	0.04746	0.04846	0.04947	0.05050	0.05155	0.05262	0.05370	0.05480	-1.6
-1.5	0.05592	0.05705	0.05821	0.05938	0.06057	0.06178	0.06301	0.06426	0.06552	0.06681	-1.5
-1.4	0.06811	0.06944	0.07078	0.07215	0.07353	0.07494	0.07636	0.07781	0.07927	0.08076	-1.4

Examples (2)


Power Function
1-beta
delta p
Power of test vs. delta p


Fitting to Gamma distribution


histogram of the difference of means
frequency
difference of means
Permutation Test of statistical differences.


Regression analysis
max temp (F)
mean temp (F)
y = 25.63+0.80x
R2 = 0.549

Process Control and Dynamics

CBE 4105, (senior year), slide materials contributed by Charles Stanier

- Prior to 2019, MATLAB was used.
- Advantages of python:
  - Integrates with other courses in Chemical Engineering curriculum and in other departments
  - Students are excited to use it
  - Pedagogical features of Jupyter notebooks (for teaching and for documenting problem solving methodology)
  - Free for download to multiple OS
  - Rapidly growing ecosystem of libraries, including in Chemical Engineering topics
  - Gives students entry point to research positions, and supports our Focus Area in Computation, Data Science, and Machine Learning

- Learning Goals by the end of the course, the student will be able to:
  - Utilize practical and mathematical tools (e.g. Python, Simulink, DeltaV, and Labview)
  - Mathematically model time-dependent real processes.

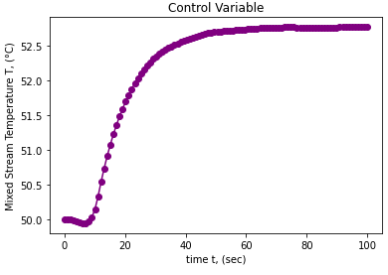
How we do it?

- **Series of coding assignments, turned in via Jupyter notebook files**
  - Students are given a resource packet (pdf) about coding, syntax, with many links to learn more
  - Students complete a progressive series of assignments
    - Hello world
    - Functions
    - Graphing
    - Dictionaries, lists, and tuples
    - Solving a single ODE
    - Solving coupled ODEs

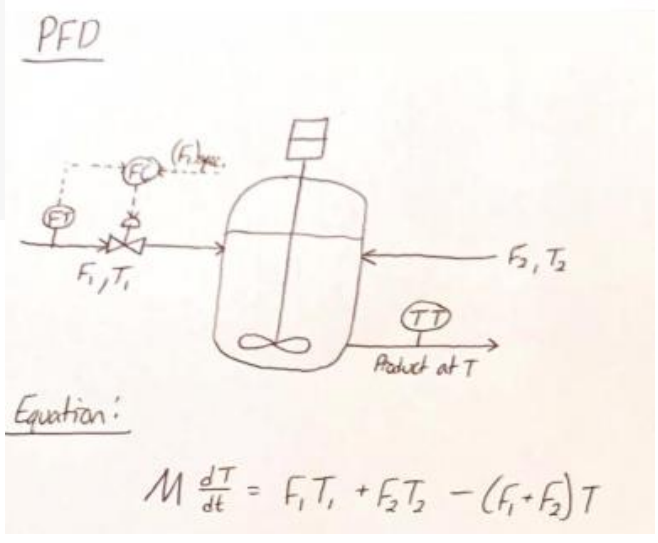
Very successful modification: In 2020, I had students choose their own system to solve. This led to higher student interest, much less copying / free-riding, more creativity, and forced students to think what kind of problems can and cannot be solved using ODE solvers

Python has been used in the required senior process control class at Iowa since 2019

In [2]:
```
# To solve a differential equation, we need a derivative file

# Defining Variables
    # M = mass of liquid in mixer (kg)
    # F1 = mass flow rate of stream 1 (kg/s)
    # F2 = mass flow rate of stream 2 (kg/s)
    # T1 = temperature of stream 1 (°C)
    # T2 = temperature of stream 2 (°C)
    # T = temperature of mixed liquid (°C)
    # t = time (s)
    # equation dTdt = ((F1*T1)/M) + ((F2*T2)/M) - (((F1+F2)*T)/M)

#Definint Derivative and setting values of variables
def deriv_func_mixer( t, T):
    PRINT_FLAG = False # change to True to print values of the derivative at each call
    M = 100
    F2 = 5
    T1 = 25
    T2 = 75

#Setting the step change
    if t>10:
        F1 = 4
    else:
        F1 = 5
```

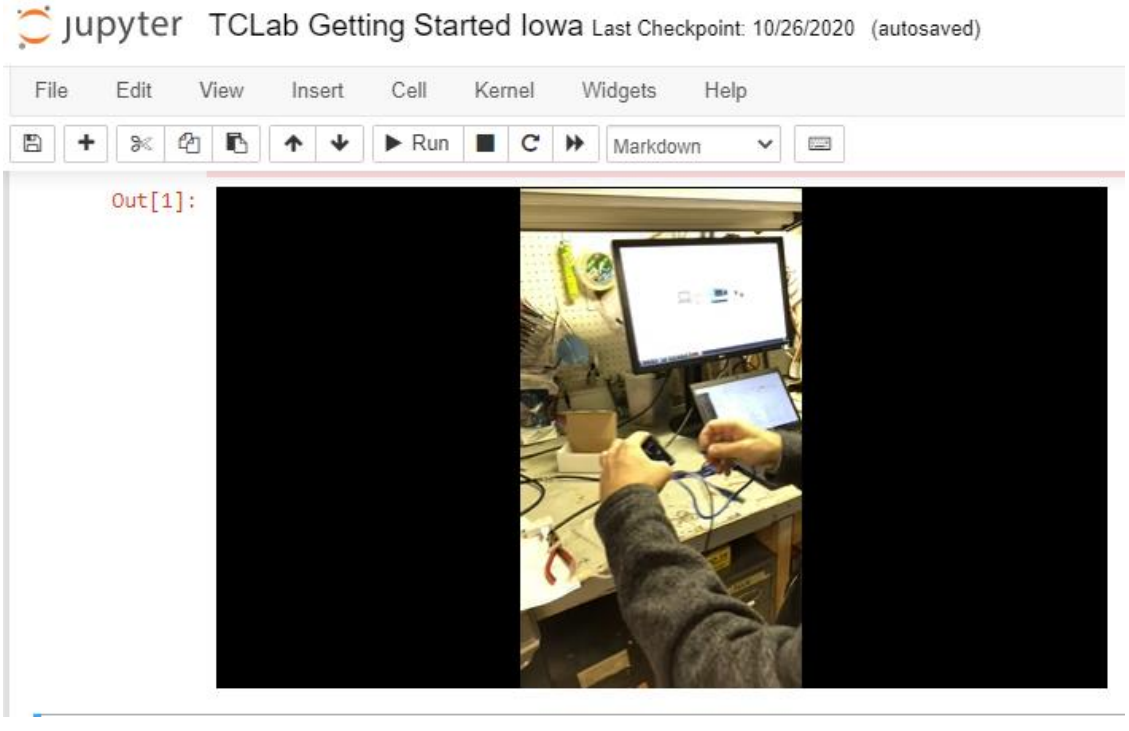


solve_ivp



A second case study from the same course

- **Use of miniature Arduino devices with sensors and actuators**



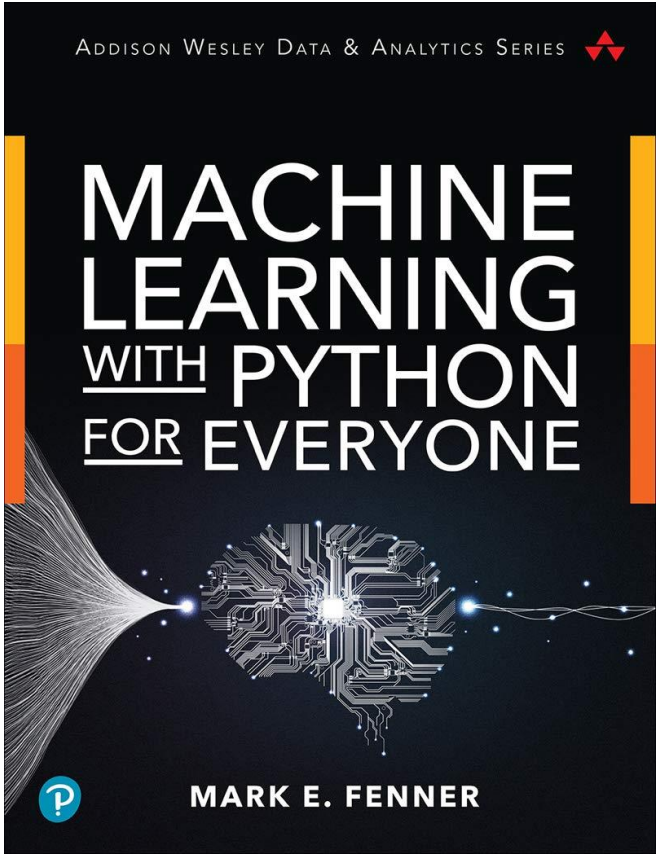
Portable Temperature Control Labs, Developed by John Hedengren, BYU *Implemented at Iowa 2018-2020*

Intro. To Machine Learning course

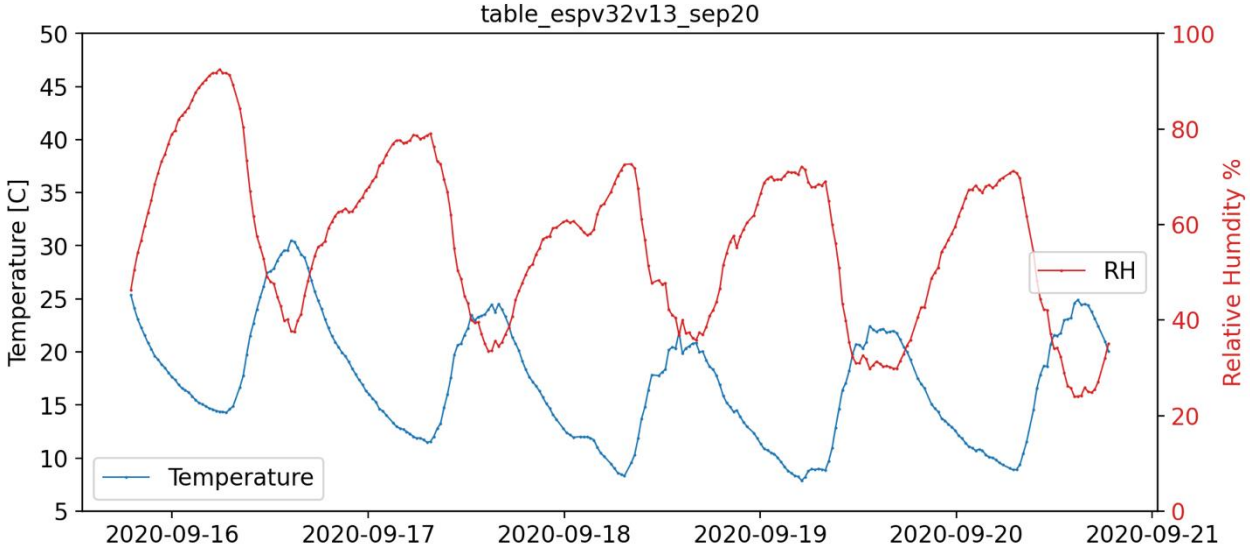
ENGR 3110: co-taught with faculty in computer engineering

Content

- Introduction/Technical Background
- Predicting Categories: Classification
- Predicting Numerical Values: Regression
- Evaluating and Comparing Learners
- Evaluating Classifiers/ & Regressors
- More Classification & Regression Methods
  - Decision trees
  - Support vector classifier
  - Regularization
  - Support vector regression
  - Regression trees
  - random forecasts
  - K-means clustering
  - PCA



Features may not be independent



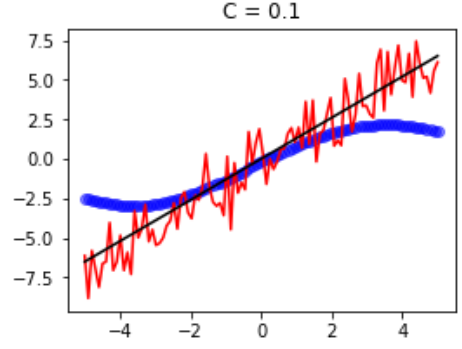
Air temperature and relative humidity at the terrace in SC 4th floor

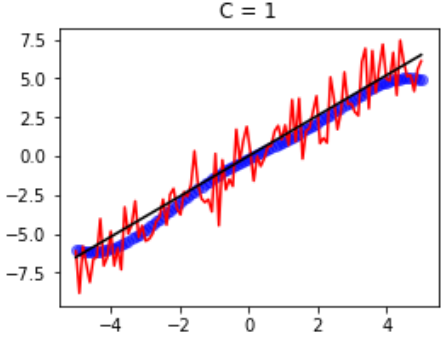
Examples (1)

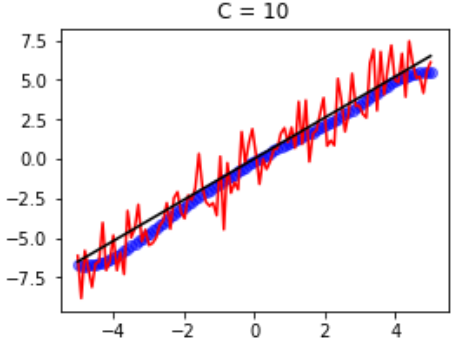
Learning the concept by trying different parameters & going beyond linear regression

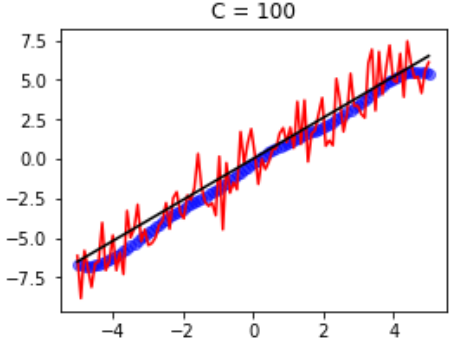
Using Supporting Vector Classifier (SVC) as an example

Bias vs. variances. The smaller C, the large bias & small variance. Large C leads to over fitting

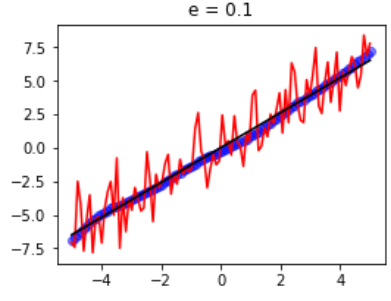


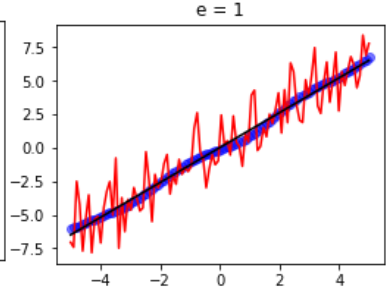


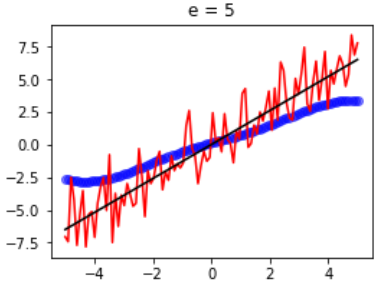


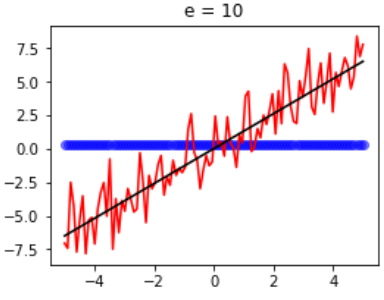


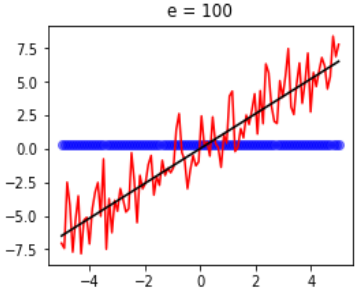
Parameters for the 'hinge' loss:







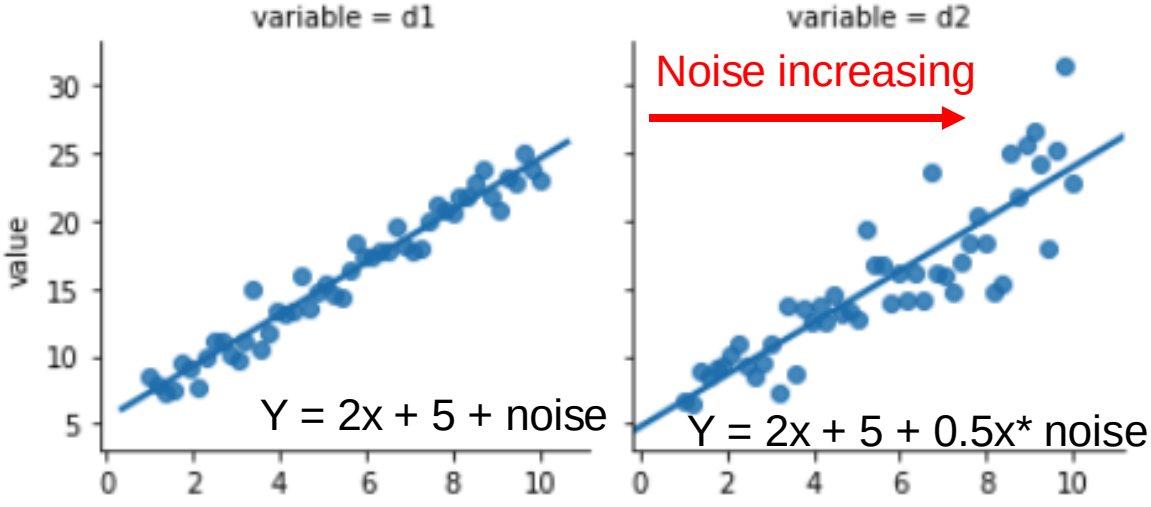




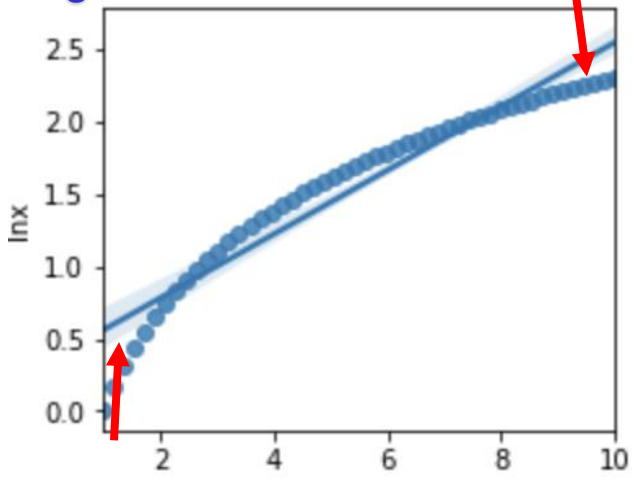
Error under which the fitting doesn't care anymore!

Examples (2)

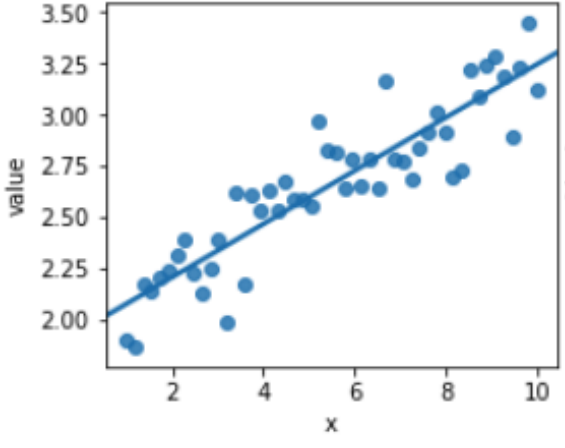
manipulate the target in the fitting

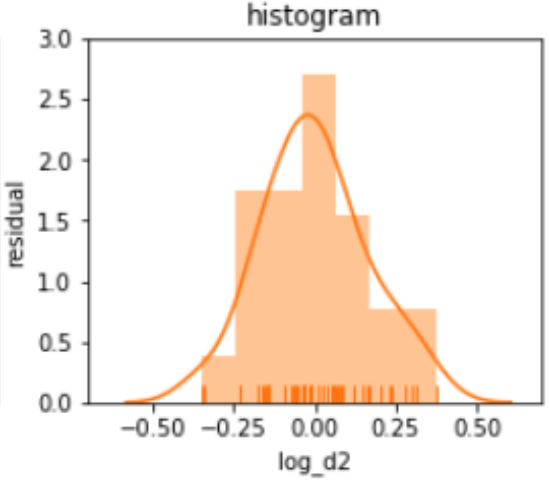


small difference



Large difference





The target is 'engineered' by log to compensate its larger noise with increase of x

Summary

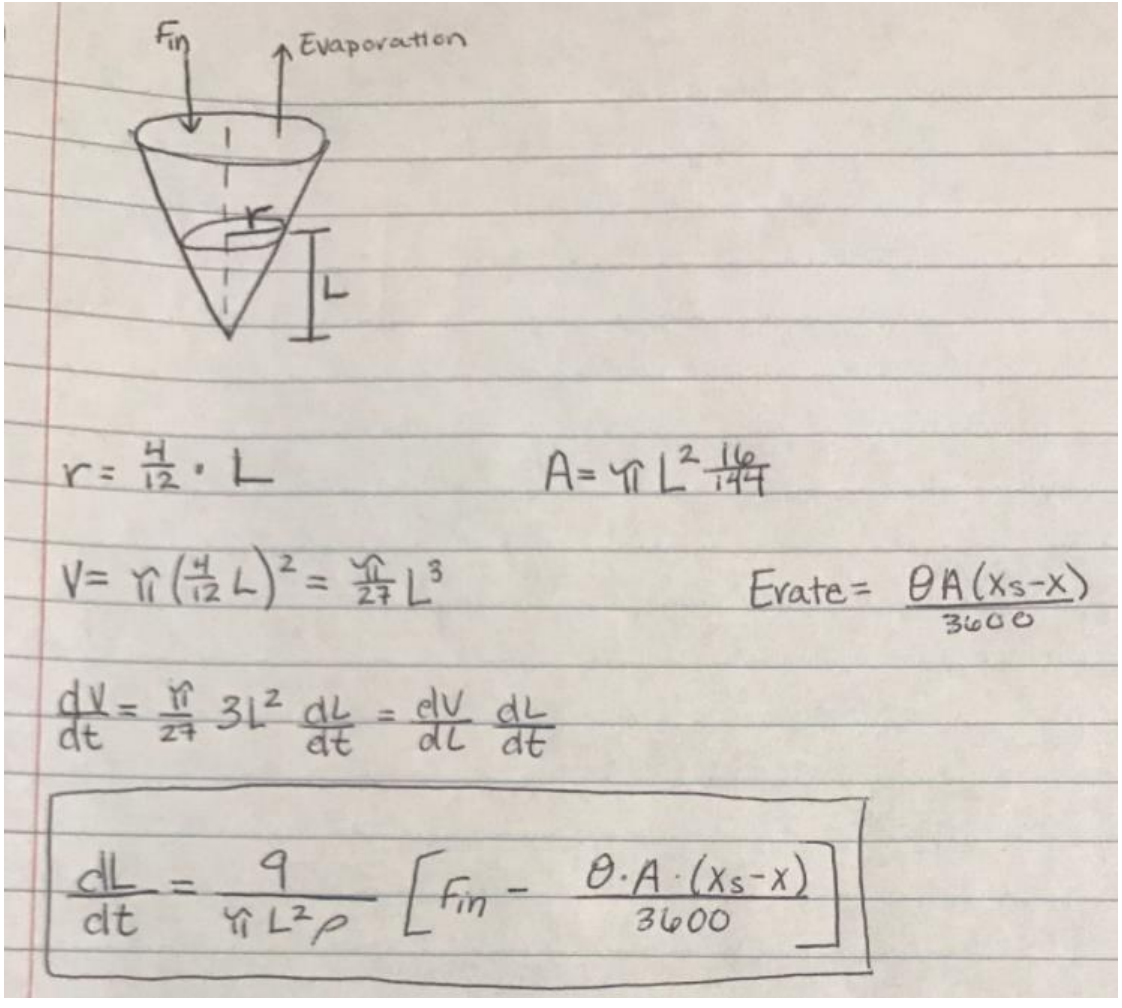
- **CBE in the Univ. of Iowa has taken concrete steps to improve and strengthen the teaching of big data science to their undergraduate students in the past 5 years:**
  - Added a numerical data analysis course that integrates classic numerical methods with modern programming/computing
  - Added a statistics course that is hands-on and programming-centric for data analysis
  - Revised process control course that now brings Python programing and modern software/sensors to incentivize students' learning of data science.
  - Co-taught the Machine Learning Intro course that is of high interest to many undergraduate students.
- **Teaching data science appears to be more effective when**
  - Projects are hands-on or have a self-design component
  - Datasets are more tangible to people's daily lives
  - Problems require the integration of theory (including statistics), numerical methods, and programming -- where rubbers meets road and fun begins!

Food for thoughts

- Teaching big data science component to undergraduates in the field of chemical engineering can be challenging because many courses might be lab experiment centric. But, the market/societal needs are often evolving.
- Chemical engineering as a discipline appears to lag behind other engineering disciplines and the fields of physical science in teaching undergraduate students about big data science and techniques.
- Most undergraduate students are equipped with analytical tools using excel, but lack training in time series analysis, multivariate statistics, and programming skills to analyze the big data – don't know how to let the computer do such data analysis jobs that are repeatable in nature.
- It is important to have a theme throughout the curriculum to ask students to solve problems from a big data perspective. How do you solve the many many problems that in nature are the same, but just having different data values for each problem?
- Recommend for senior design project to have a big data component (if all possible).
- Having students debug the codes together as a group can be fun and be equivalent to the team project or team lab experiment.

Please join us.

Even a simpler problem, designed and solved by the student, still demonstrated the skill



$r = \frac{4}{12} \cdot L$

$A = \pi L^2 \frac{16}{144}$

$V = \pi \left(\frac{4}{12} L\right)^2 = \frac{\pi}{27} L^3$

$Erate = \frac{\theta A (X_s - X)}{3600}$

$\frac{dV}{dt} = \frac{\pi}{27} 3L^2 \frac{dL}{dt} = \frac{dV}{dL} \frac{dL}{dt}$

$$\frac{dL}{dt} = \frac{9}{\pi L^2 \rho} \left[F_{in} - \frac{\theta \cdot A \cdot (X_s - X)}{3600} \right]$$

All from student's Jupyter notebook

Variables

Variable	Symbol	Value	Units
flow rate in	Fin	0.005	kg/sec
radius	r	see equation	meters
height	L	initial = 0.5	meters
density of water	ρ	997	kg/L

Check

$$0 = \frac{9}{\pi L^2 (997)} \left[0.005 - \frac{(34.5)\left(\pi \cdot \frac{16}{149} \cdot L^2\right)(.02 - 0.0079)}{3600} \right]$$

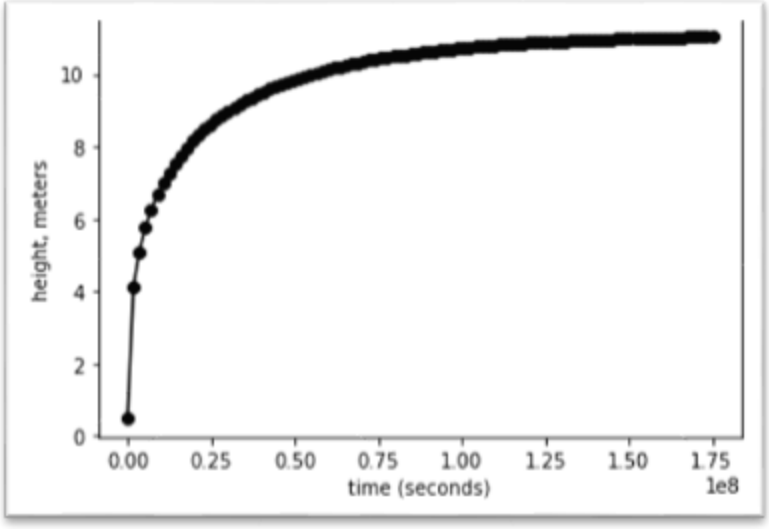
$$0 = \frac{0.00287}{L^2} \left(0.005 - 4.05 \cdot 10^{-5} L^2 \right)$$

$$0 = \frac{1.437 \cdot 10^{-5}}{L^2} - 1.163 \cdot 10^{-7}$$

$$1.163 \cdot 10^{-7} L^2 = 1.437 \cdot 10^{-5}$$

$$L^2 = 123.53$$

$$L = 11.11 \text{ m}$$



D. Last step, but very important -- comment on the solution.

The model behaves as predicted. It starts at the initial height of 0.5 meters of water. At the time of forcing there is a sharp increase that then begins to flatten out into steady state.

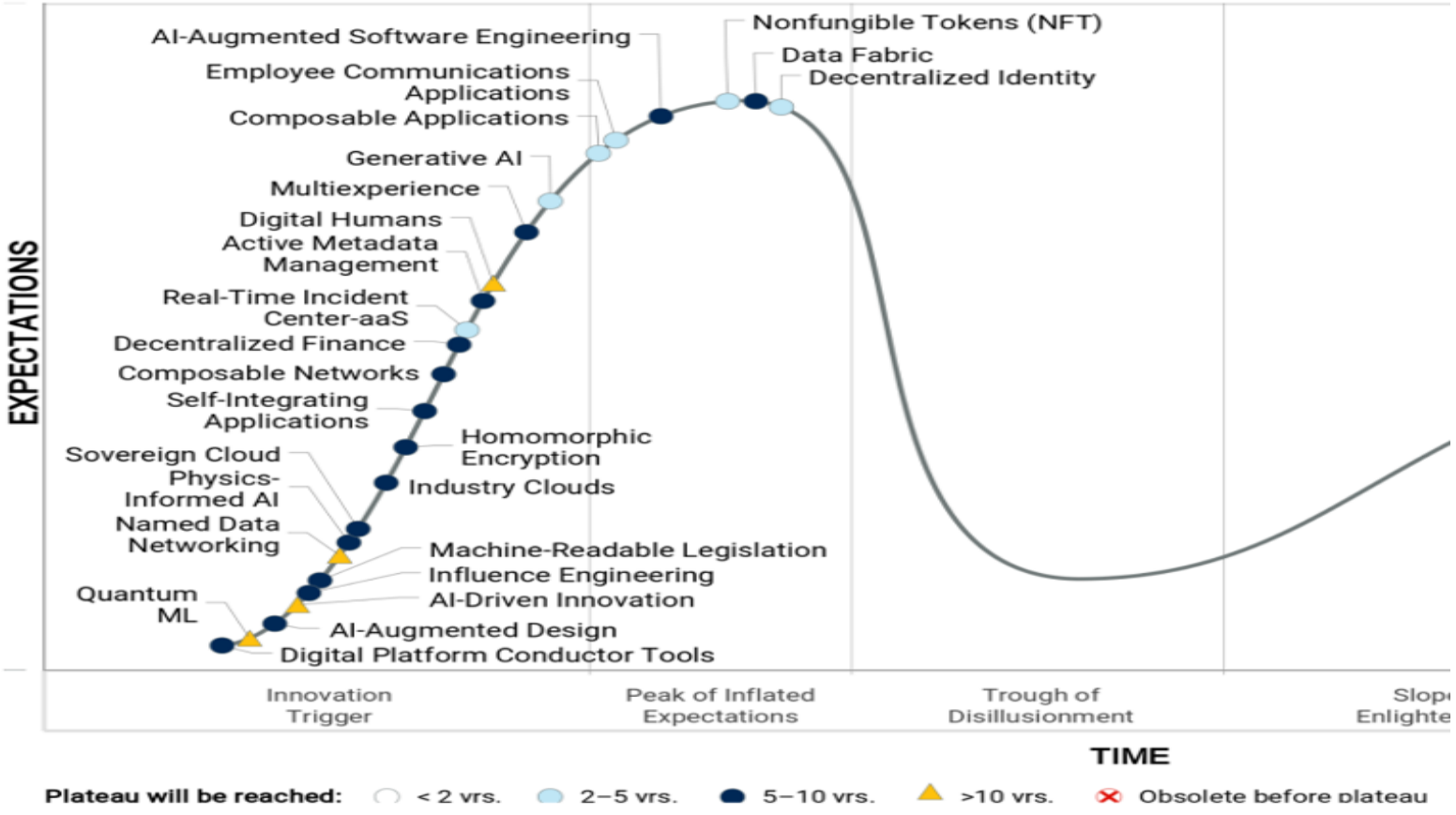
Potential Uses

Potential Teaching and Learning Use	At Iowa?
Teach coding (by having a device to interface with, sending commands, receiving data)	
Teaching dynamics, heat transfer, model fitting	X
Teaching PID process control	X
Teaching advanced process control	
Individual lab for remote learning, or in cases where pilot-scale controls labs are not available	X
Prelab to prepare students for pilot-scale control labs	X

Main drawbacks

- While python is OS independent, the device drivers are device independent. Expect to have some troubleshooting and some failures of communication if students use on their own devices.
- GUI options for PID control require several dependencies and there were some glitches that we were not able to fix in all cases.

Hype Cycle for Emerging Technologies, 2021



Source: Gartner (August 2021)

747576


Hawkeyes! UIOWA!
The world's most valuable resource
Data and the new rules of competition
Google
amazon
Microsoft
Google
UBER
TESLA
TESLA

A bit intro about me

background & teaching experience

- First experience with computer programming
  - BASIC, 1989, first year in high school. But never used after learning it for one semester.
- Learned Fortran 77 at the sophomore year and have used Fortran for computing & research since then (1993)
  - It was a required course for students majoring in meteorology/atmospheric sciences
  - Used throughout the undergraduate education because many courses (including project design) builds upon the weather data and computing.
  - Final design was to develop a simple two-layer atmospheric model to solve the isentropic dynamic equations on a Cray machine to predict large-scale weather.
  - Great learning experience via team work. Fond memories of working together to resolve the bugs in the code.
- 17 years of teaching in two universities (self-taught Python through the years)
  - University of Nebraska – Lincoln; students majoring in atmospheric sciences are introduced by programming at the sophomore year and are trained on data analysis with programming throughout the curriculum.
  - University of Iowa; students majoring in chemical engineering are introduced to programming in freshman year via ”problem solving”…

Intro. to Problem Solving

ENGR 1100 in U. Iowa

- Development and demonstration of specific problem-solving skills; directed project or case study involving actual engineering problems and their solutions.
- **Module 3: Technical Representation, Presentation, and Analysis of Data**
  - Introduction to Excel and MATLAB for matrix solving (and some statistical analysis)
  - Solve simultaneous equations in matrix form using Excel and MATLAB
  - Parameter estimation methods: Method of selected points and least-squares regression
  - Application of graphing and data analysis
  - Non-linear analysis, curve fitting

Intro to Engr Computing

ENGR 1300 in U. Iowa

- Engineering problem solving using computers; introduction to digital computations, problem formulation using a procedural high-level language; structured, top-down program design methodology; debugging and testing; introduction to use of software libraries; examples from numerical analysis and contemporary applications in engineering.

Process Calculation
aka: material and energy balance

CBE 2150 in U. Iowa

- This course provides an overview of the fundamentals of chemical engineering. Students learn and practice problem-solving approaches to solve material and energy balances for chemical processes with a focus on steady-state systems. In the context of practical applications from various industries, balances around commonly used unit operations are written and solved with consideration of chemical reactions, phases, and process variables such as temperature and pressure.