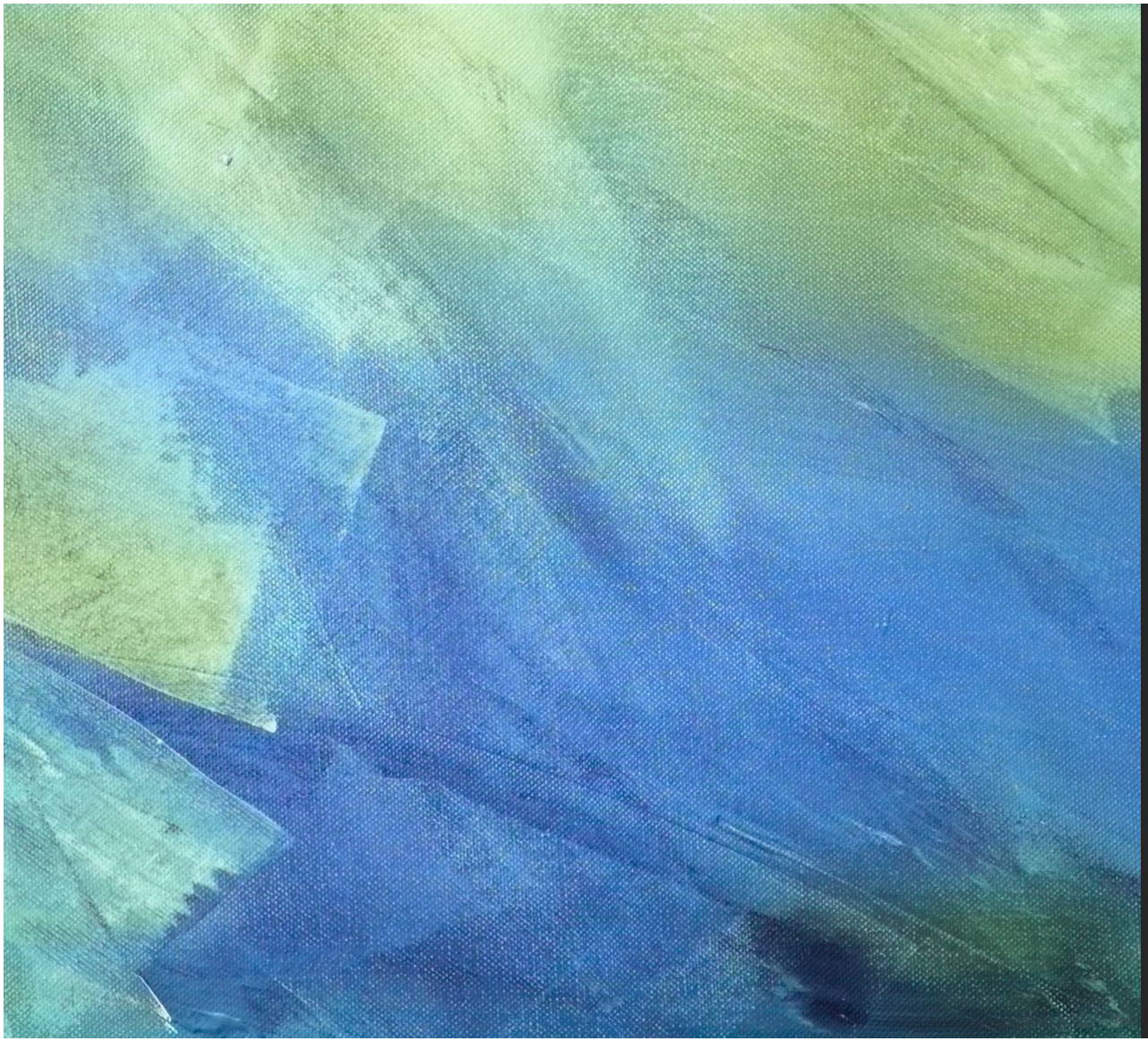


DATA TECHNIQUES FOR ENGINEERS AND SCIENTISTS

J.D. LANDGREBE

DEPT. OF CHEMICAL ENGINEERING

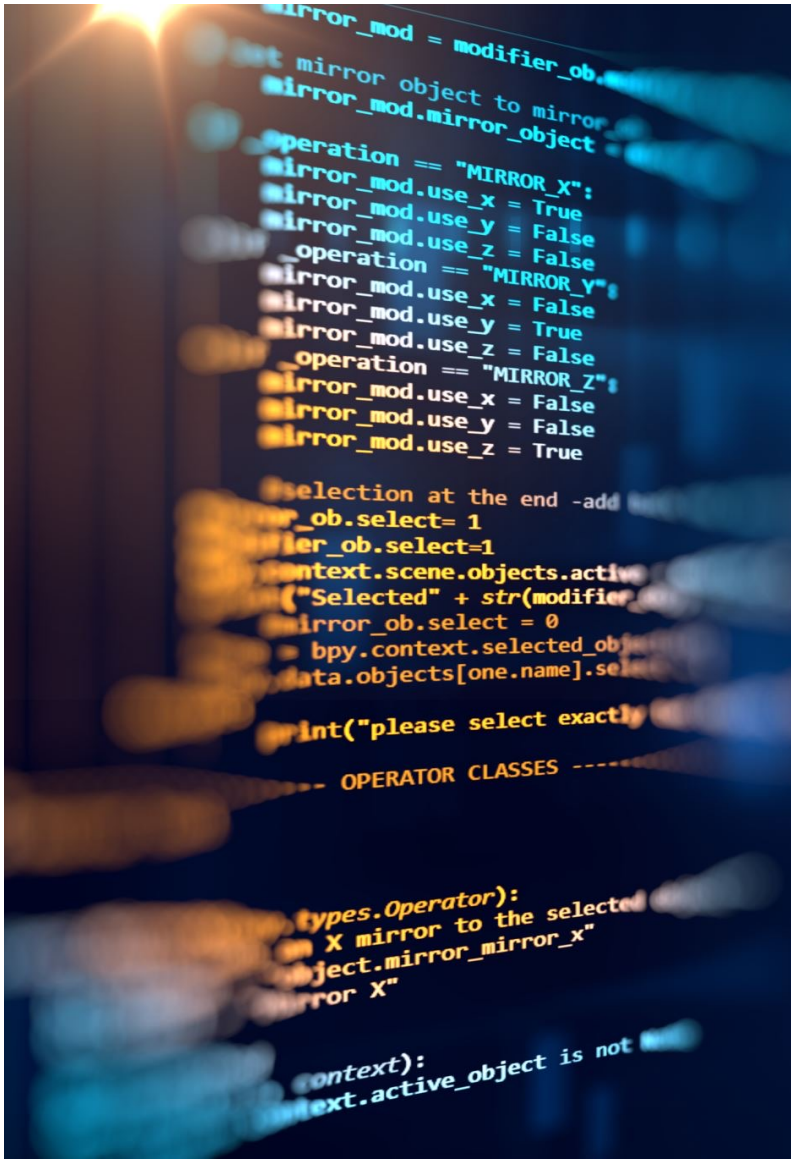
UNIVERSITY OF DELAWARE

LECTURE OUTLINE TEMP

- Baseline Software for engineering work – data analysis and modeling
- Exercise: install Anaconda/run intro to Pandas
- Understand available Python packages for general work (and understand Anaconda interaction with them)
- Nomenclature and data architecture for experiments
- What is experiment Curation?
- Data reshaping techniques Joining and summarizing an experiment –
- Exercise with JMP Tables menu – Join, Concatenate
- Parsing raw data – Exercise
- Work a case study of analyzing an experiment – TBD based on Will input

SOFTWARE FOR ENGINEERS

- Pick software by “use case” and use each software in its “lane” where it is best tool for the job
- “Use cases” is a useful way of compartmentalizing tasks or activities in software
- Use cases start with “I need to…” (or “my customer needs to…”)
  - My supervisor needs to be able to scroll through our raw data to understand the results
  - I need to create a correlation matrix for my raw material’s quality data
  - I need to build a validated calculation model for reaction kinetics
- Use case focus
  - Avoids getting (yourself and others) distracted by things that software X is not good at – there is no perfect software tool
  - Puts a premium on making data portable across applications



GENERALIST ENGINEER/SCIENTIST SOFTWARE RECOS BY USE CASE

- **JMP® software** - design of experiments (DOE), exploratory analysis and data visualization with accompanying statistical analysis
- **Microsoft Excel®** - make data usable by others, generate “end of the pipeline” reports and create spreadsheet models with calculations for use by yourself or non-coders
- **Python scripts** for data reshaping and for developing coded models and data pipelines (possibly mixing in a little SQL)

Blog

The Right Modeling And Analysis Tools For the Job

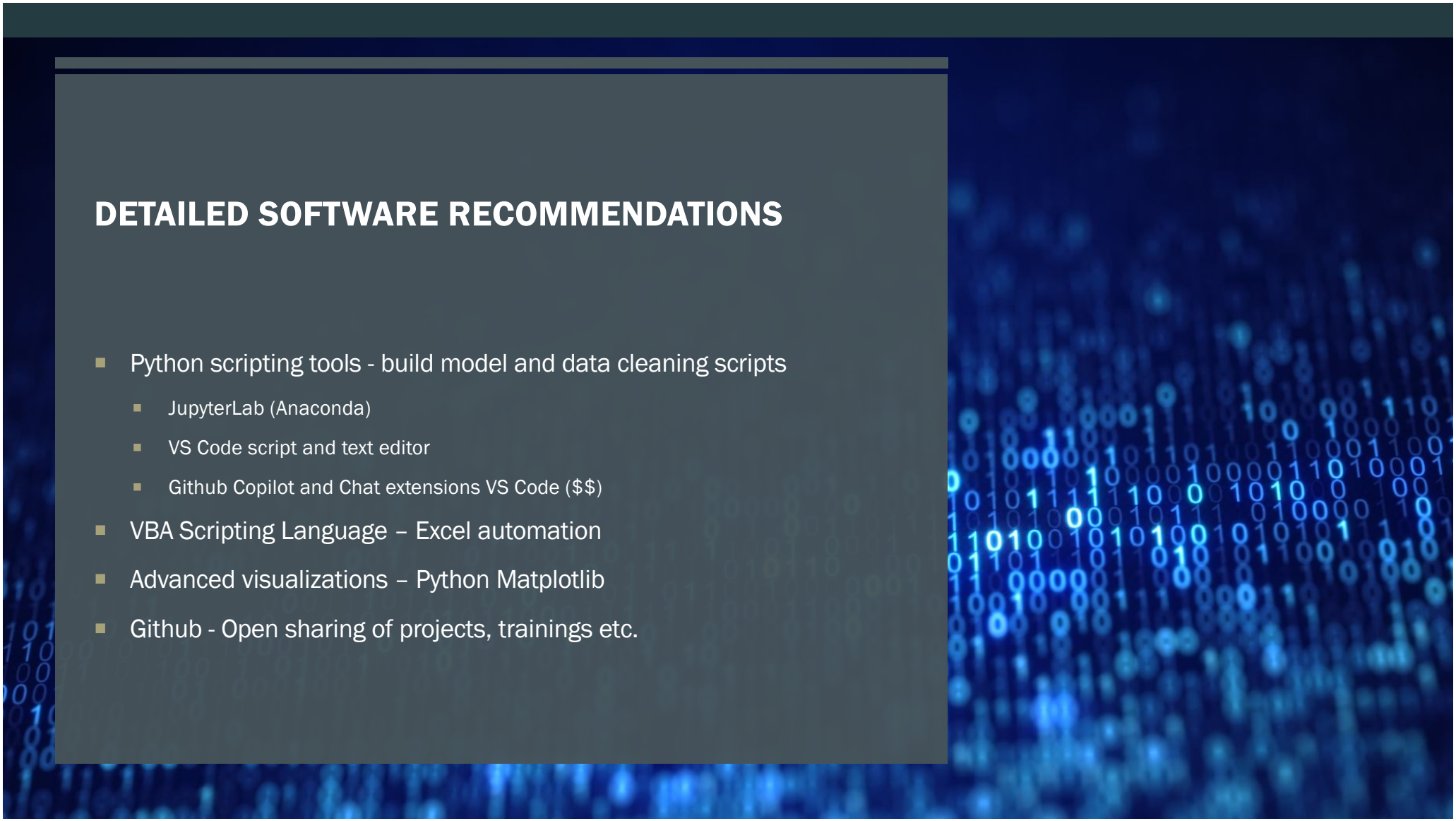
Software tools in the data and modeling arena often lead individuals and teams into counterproductive patterns. By being informed and intentional, you can choose the best tool for a particular job. It's good to recognize that software providers, meaning companies and open-source communities, keep...

READ MORE

https://datadelveengineer.com/the-right-modeling-and-analysis-tools-for-the-job/

DETAILED SOFTWARE RECOMMENDATIONS

- Python scripting tools - build model and data cleaning scripts
  - JupyterLab (Anaconda)
  - VS Code script and text editor
  - Github Copilot and Chat extensions VS Code ($$)
- VBA Scripting Language – Excel automation
- Advanced visualizations – Python Matplotlib
- Github - Open sharing of projects, trainings etc.



PYTHON LIBRARIES TO EXTEND FUNCTIONALITY

Useful Python libraries for Engineers and Scientists

You can get these with Anaconda installation!!

- Pandas – Data reshaping and analysis
- Numpy – Numerical calculations
- Pytest – Test/Validate your scripts
- OpenPyXL – Format outputs in Excel
- Re (Regular Expressions aka Regex) – Parse text strings

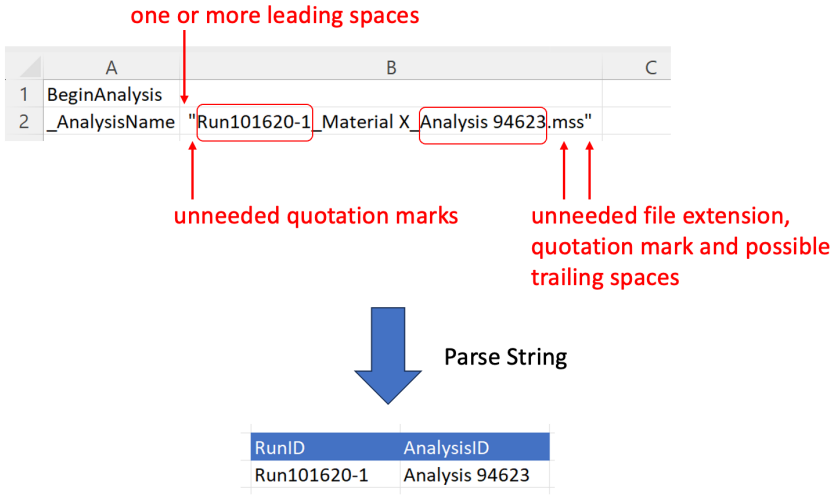
EXAMPLE: PYTHON REGEX PACKAGE TO EXTRACT STRINGS

regex 2023.10.3

Lab instrument software outputs IDs in string needing cleanup

regex Python package can strip out unneeded quotation marks, spaces and (not shown) file extension

(Not shown) Either regex or Python split() command can break the string into pieces using underscore characters as delimiters



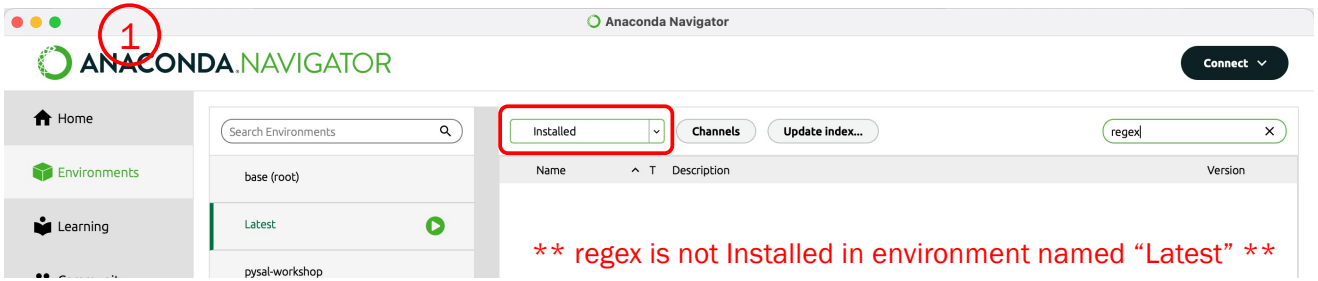
libs > curve_parse.py > ...

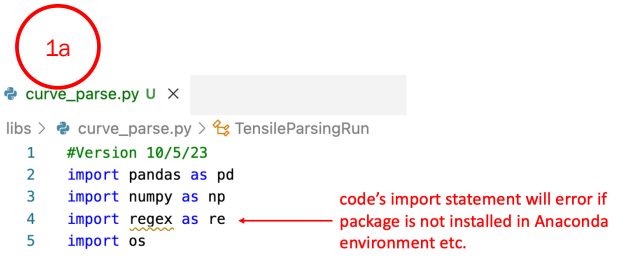
```
1   #Version 10/5/23
2   import pandas as pd
3   import numpy as np
4   import regex as re
5   import os
```

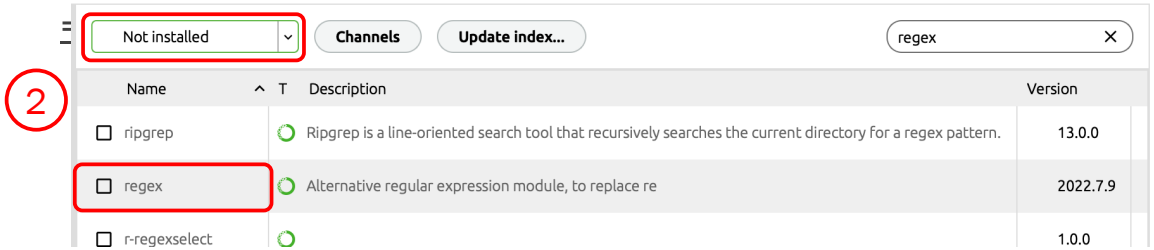
```
174     def id_string_cleanup(self, id_string):
175         """
176         Strip leading/trailing spaces and quotes from an ID string
177         JDL 10/5/23
178         """
179         # Define regular expression patterns
180         leading_space_pattern = r'^\s+'
181         trailing_space_pattern = r'\s+$'
182         quote_pattern = r'^"|"$'
183
184         # Use regular expression to strip leading/trailing spaces and quotes
185         id_string = re.sub(leading_space_pattern, '', id_string)
186         id_string = re.sub(trailing_space_pattern, '', id_string)
187         id_string = re.sub(quote_pattern, '', id_string)
188         return id_string
```

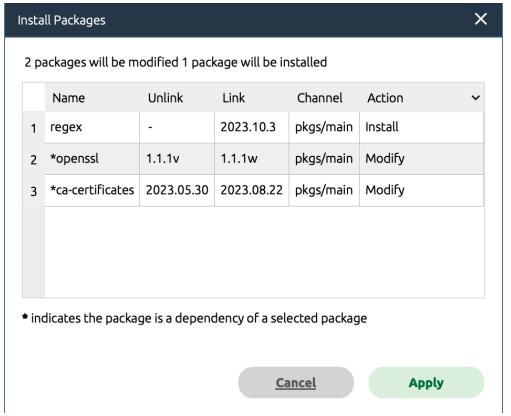
Regex code for "zero or more spaces at beginning of string" ("ChatGPT pls help!!")

INSTALLING PACKAGES (AKA LIBRARIES) IN ANACONDA









Anaconda selects related (aka “dependent”) packages needed by the selected package

EXERCISE WITH PANDAS OR PYTHON REGEX

Exercise 1

- Have students ask ChatGPT for a regex to parse a particular string
- Have students try it in ChatGPT or Copilot
- Try their result on some data

Exercise 2

- Ask students/groups to clone Intro_To_Pandas_For_Noncoders
- Run notebook with imported data
- Ask summary question(s) – How many rows in x subset?

OBJECTIVES
WORKING WITH EXPERIMENT DATA

Experiments* have a common data structure (we will call it “architecture”).

Recognizing this lets us master common data techniques to analyze efficiently and have data in good formats for graphing

Objectives

- Learn reusable terms for the data elements from experiments
- Learn how use software to do needed transformations to go from raw data to analyzed summary
- Learn how to “curate” experimental data to make it easy to find and share

* Data don’t know whether they were generated in a lab or virtual experiment, so this applies to computer modeling data in addition to physical experiments

WHAT ARE EXPERIMENTS?

Experiments Cover An amazing range for engineers and scientists!

Make a production or lab-scale batch of several product formulations

Run a packing line using pre-set conditions for a set amount of time

Crash a sensor-equipped car into a wall

Diaper a baby with several product designs

Drive a F1 car around a track with several various fuel mixtures

Ask a man to shave his face with 3 different types of razors on consecutive days

Make batches of cookies with different types of chocolate

Wash test, fabric swatches with different detergent formulations

Use different catalysts to run a chemical conversion with the same starting materials

Use PCR to replicate and sequence the DNA from multiple virus samples

Mop floors with mops using different cleaning solutions

Surgically implant artificial joints manufactured with different polymeric coatings

WHAT DO THESE HAVE IN COMMON?

Experiments (or ”**studies**”…we will use synonymously) are a building block for designing and confirming products.

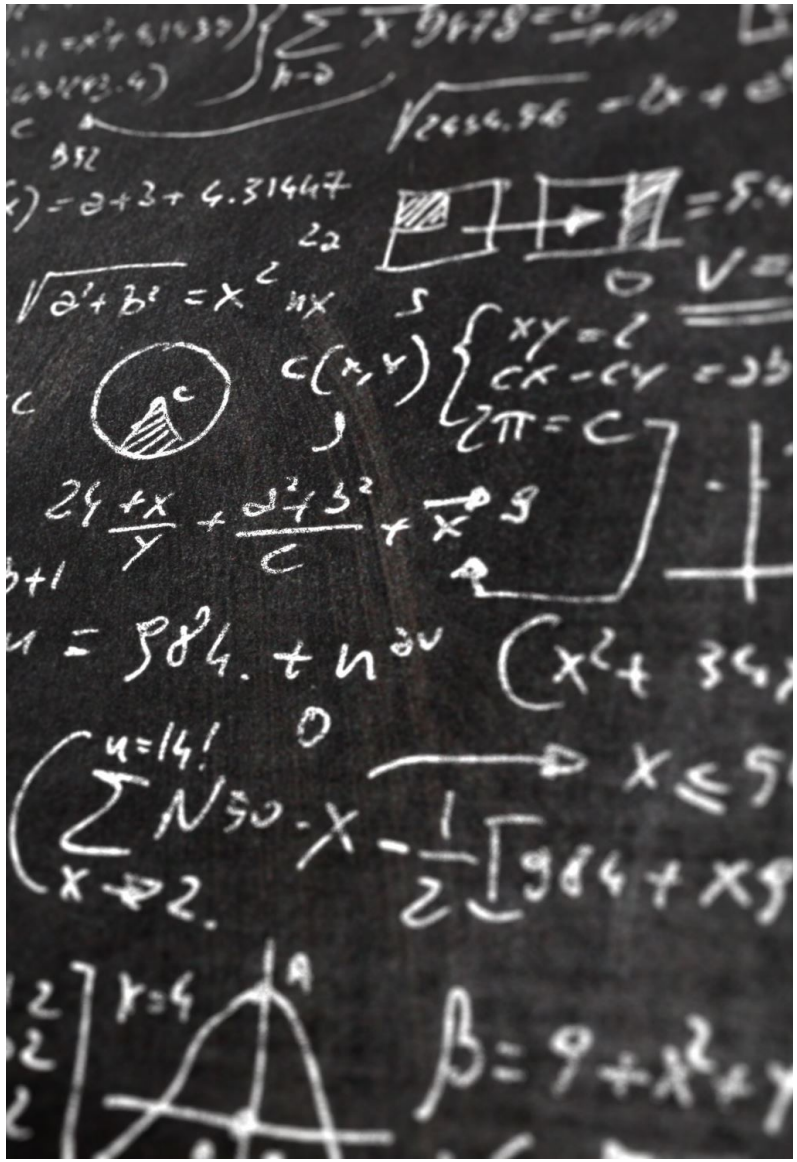
Experiments have “**runs**” consisting of one or more conditions having controlled inputs such as the starting materials and test conditions

A run can be described by a list of the “**run variable**” values which are the pre-set inputs and important ambient conditions recorded during the run.

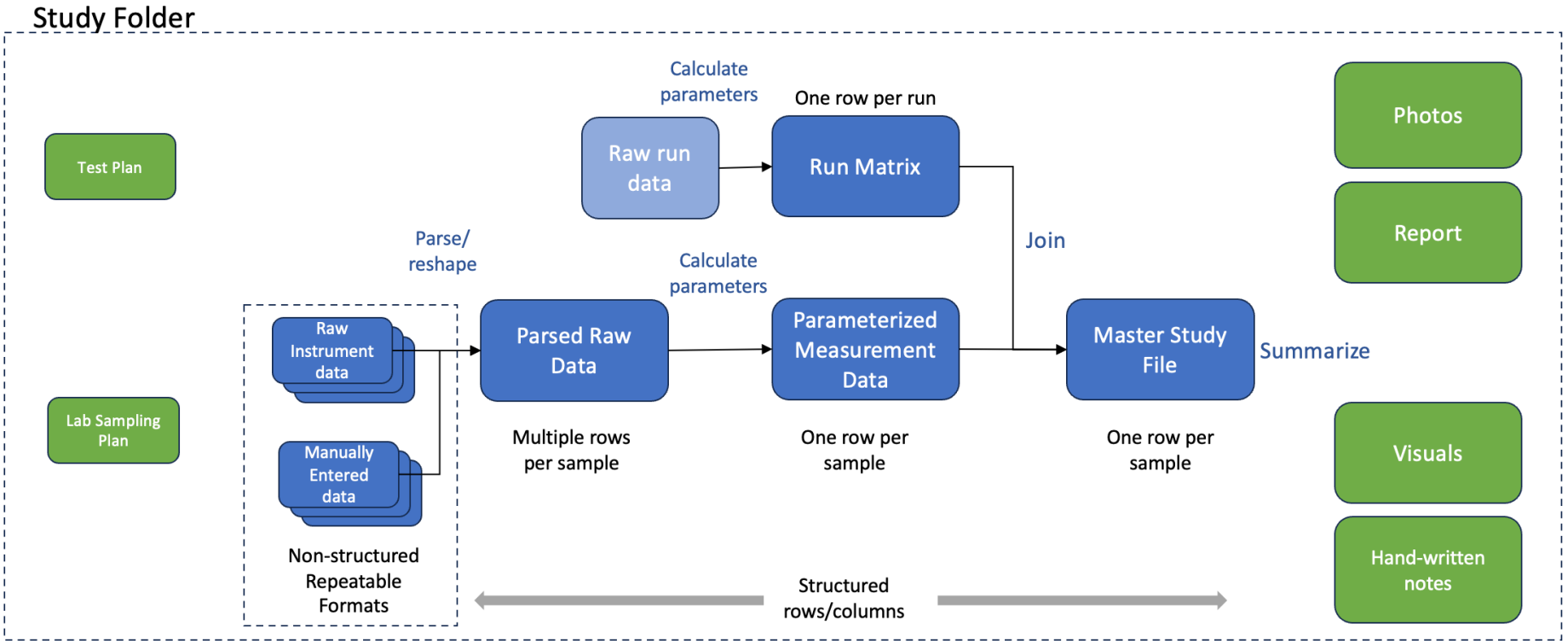
Experiments generate “**raw data**” consisting of the original measurements to assess what happened (detailed speed versus time data for F1 car circuit around a track)

The raw data gets converted to “**parameterized data**” which is quantities reflecting the outcome (average speed for a complete track circuit)

Run variable values can be assembled into a “**run matrix**” with a single row of values for each run

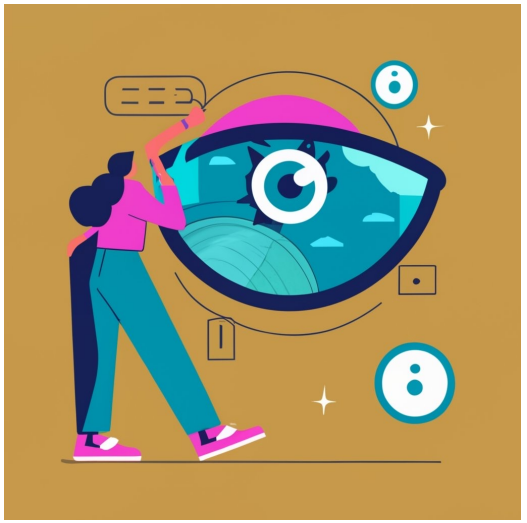


EXPERIMENT DATA "ARCHITECTURE"



WHY DOES STANDARD STUDY NOMENCLATURE MATTER?

- Reusable ”lens” that you can use in any context
- Knowing nomenclature and architecture gives you a “seeing eye” for how to plan and organize experiments
- Identifies efficient, generic ways of converting the Run Matrix and Raw Data into analyzed data and conclusions
  - Parsers to convert raw data to structured tables
  - “Join” and “Concatenate” to bring data together into a “Master Study File”



EXERCISE WITH JMP

Data Techniques exercise with JMP tables menu for reshaping typical data – Join, Concatenate, Stack, Split

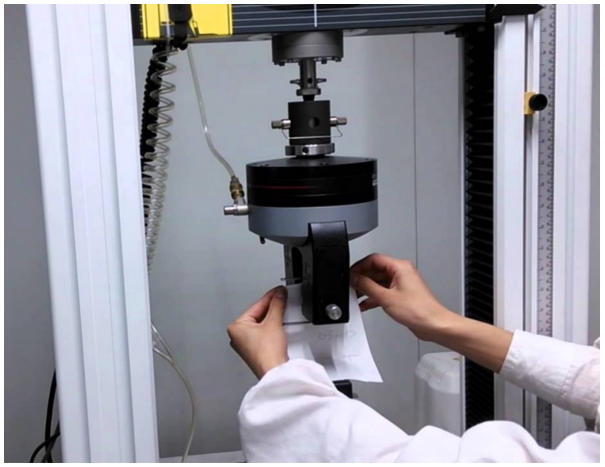
- Break into groups
- Ask them to work the exercises individually + together and compare notes
- Study questions: Share with your group an interesting dialog box option you found
- JDL demo

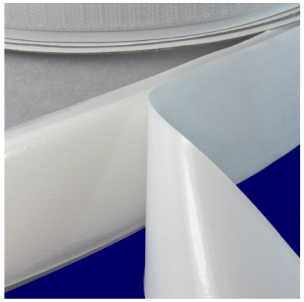
PARSING RAW DATA

- “Parsing” refers to going from a repeatable non-structured format to structured rows/columns
- Parsing needs to identify samples by its RunID, SampleID (e.g. sampling time/location) and AnalysisID metadata for the result to be considered “structured” (e.g. each sample’s data uniquely identified and traceable back to its source)
- **Study owner needs to pre-plan/intervene with lab to ensure metadata are available/parseable post-measurement!!!**
- Parsing requirements vary [wildly!] depending on data source
- Manually-entered data are a special case because humans are very creative about how they choose to enter data
- Lab instruments typically have specialized software that outputs raw and/or parameterized data

RAW DATA PARSING EXAMPLE
DATA FROM MTS TENSILE TESTER SOFTWARE

- Testing involves pulling apart substrate(s) such as hook and loop fastener or adhesive-coated strip + landing surface
- Measurement of force versus displacement while separating the materials –peak load and average load are key results
- Requires multiple samples to get good data: Testing is noisy due to sample prep of joining substrates and general noise from failure testing



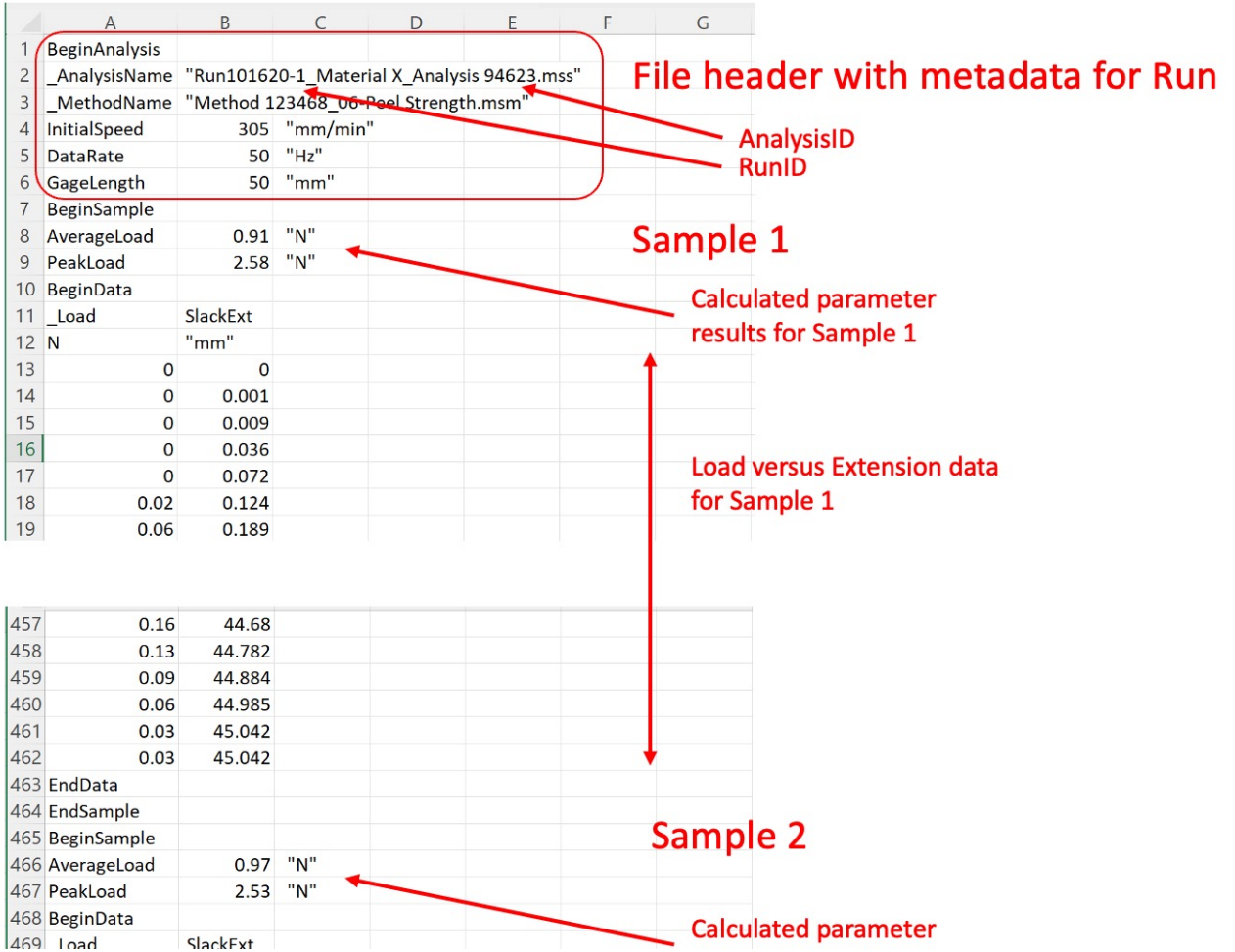


DuraGrip® Brand Adhesive
Backed Hook and Loop Fasteners

VELCRO® Brand Adhesive
Backed Hook and Loop Fasteners

RAW DATA PARSING EXAMPLE DATA FROM MTS TENSILE TESTER SOFTWARE

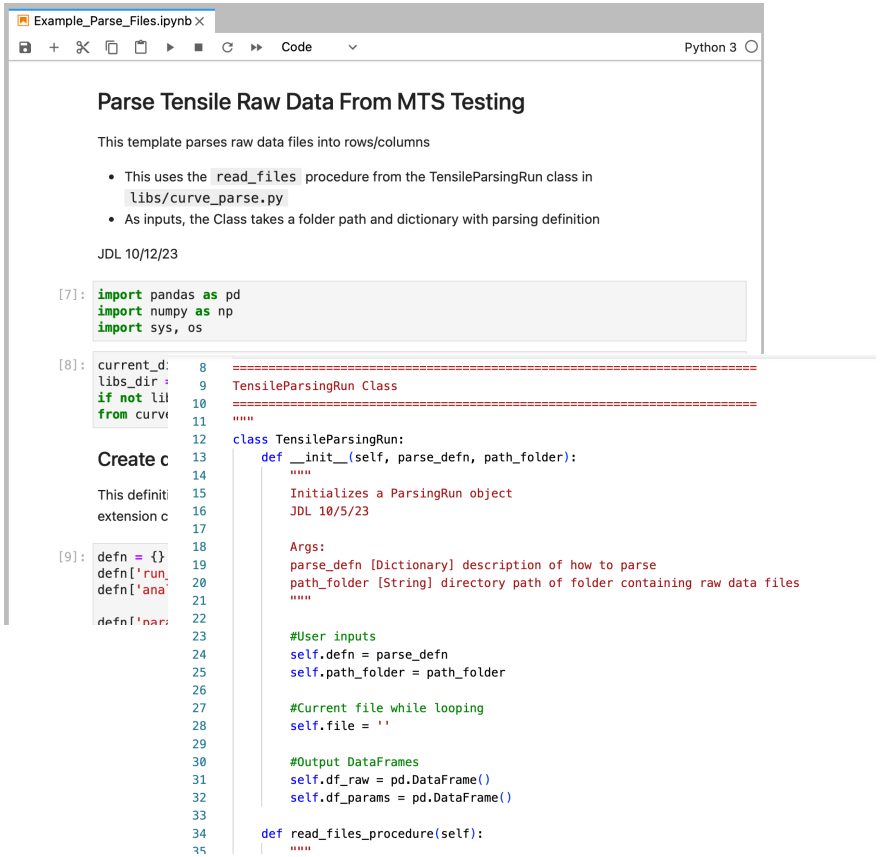
- Instrument software outputs files containing data for multiple samples from a run
- Raw files contain a combination of calculated parameters and raw data –we want to analyze both!!



	A	B	C	D	E	F	G
1	BeginAnalysis						
2	_AnalysisName	"Run101620-1_Material X_Analysis 94623.mss"					
3	_MethodName	"Method 123468_06-Peel Strength.msm"					
4	InitialSpeed	305	"mm/min"				
5	DataRate	50	"Hz"				
6	GageLength	50	"mm"				
7	BeginSample						
8	AverageLoad	0.91	"N"				
9	PeakLoad	2.58	"N"				
10	BeginData						
11	_Load	SlackExt					
12	N	"mm"					
13	0	0					
14	0	0.001					
15	0	0.009					
16	0	0.036					
17	0	0.072					
18	0.02	0.124					
19	0.06	0.189					
457	0.16	44.68					
458	0.13	44.782					
459	0.09	44.884					
460	0.06	44.985					
461	0.03	45.042					
462	0.03	45.042					
463	EndData						
464	EndSample						
465	BeginSample						
466	AverageLoad	0.97	"N"				
467	PeakLoad	2.53	"N"				
468	BeginData						
469	_Load	SlackExt					

CUSTOM PYTHON PARSER FOR TENSILE DATA

- To try it with example data, clone from: https://github.com/jlandgre/Python_Curve_Parser
- Parser consists of a Python Class in curve_parse.py (run with example data from Example_Parse_Files.ipynb)
- Creates output files: df_params.xlsx and df_raw.xlsx



JMP ANALYSIS OF PARSED DATA

- xxx
- Run101620-2 has significantly higher Peak Load than -1 ($p<0.05$)
- yyy