Advanced Data Management (CSCI 640/490)

Scalable Dataframes

Dr. David Koop

History of Dataframes

- Originally in *Statistical Models in S*, [J. M. Chambers & T. J. Hastie, 1992]
- R, open-source alternative to S, developed in 2000 (with dataframes)
- Pandas, 2009
- Spark, 2010 (resilient distributed dataset [RDD], Dataset API)

[D. Petersohn, 2022]

Pandas Workflow: Ingest, Cleaning, Analysis

R1. Read HTML

```
import pandas as pd
products = pd.read_html( ... )
products
```

	iPhone 11 Pro	iPhone Pro Max	iPhone 11	...
Display	5.8-inch	6.5-inch	6.1-inch	...
Camera	Triple 12MP	Triple 12MP	Dual 12MP	...
Front Camera	120MP	12MP	7MP	...
...	...	...	...	...

C1. Ordered point updates

```
products.iloc[2, 0] = "12MP"
products
```

	iPhone 11 Pro	iPhone Pro Max	iPhone 11	...
Display	5.8-inch	6.5-inch	6.1-inch	...
Camera	Triple 12MP	Triple 12MP	Dual 12MP	...
Front Camera	12MP	12MP	7MP	...
...	...	...	...	...

C2. Matrix-like transpose

```
products = products.T
products
```

	Display	Camera	...	Wireless Charging
iPhone 11 Pro	5.8-inch	Triple 12MP	...	Yes
iPhone Pro Max	6.5-inch	Triple 12MP	...	Yes
iPhone 11	6.1-inch	Dual 12MP	...	Yes
iPhone XS	5.8-inch	Dual 12MP	...	No

C3. Column transformation

```
products = products\
 ["Wireless Charging"].map(
  lambda x: 1 if x is "Yes" else 0)
products
```

	Display	Camera	...	Wireless Charging
iPhone 11 Pro	5.8-inch	Triple 12MP	...	1
iPhone Pro Max	6.5-inch	Triple 12MP	...	1
iPhone 11	6.1-inch	Dual 12MP	...	1
iPhone XS	5.8-inch	Dual 12MP	...	0

C4. Read Excel

```
prices = pd.read_excel( ... )
prices
```

	Price	Rating
iPhone 11 Pro	999.00	4.5
iPhone Pro Max	1099.00	5.0
iPhone 11	699.99	4.6
iPhone XS	999.99	4.7

A1. One-to-many column mapping **A2. Joins**

```
one_hot_df = pd.get_dummies(products)
iphone_df = prices.merge(
    one_hot_df,
    left_index=True, right_index=True
)
iphone_df
```

	Price	Rating	Wireless Charging	Display_\ 5.8-inch	...
iPhone 11 Pro	999.00	4.5	1	1	...
iPhone Pro Max	1099.00	5.0	1	0	...
iPhone 11	699.99	4.6	1	0	...
iPhone XS	999.99	4.7	0	1	...

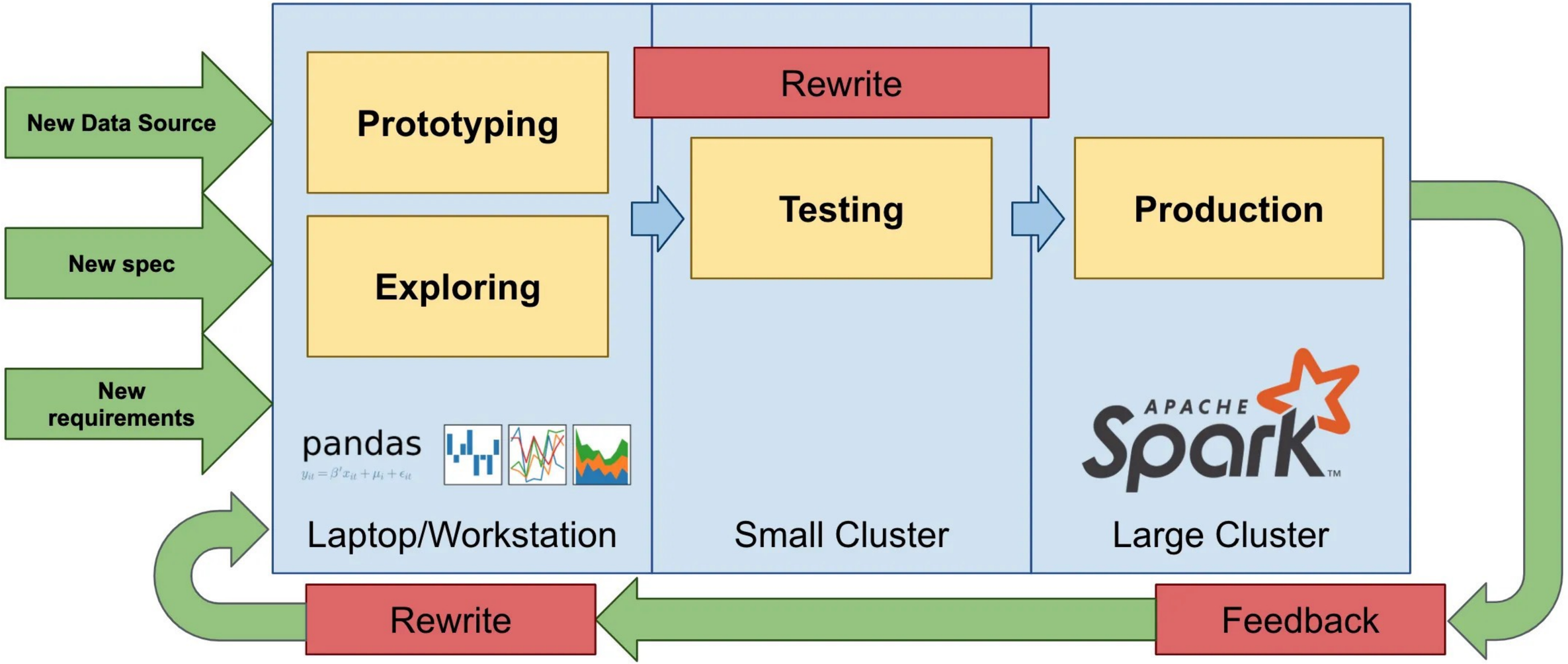
A3. Matrix Covariance

```
iphone_df.cov()
iphone_df
```

	Price	Rating	...
Price	29868.3	19.967	...
Rating	19.967	0.0466667	...
Wireless Charging	-16.8317	-7.40149e-17	...
Display_5.8-inch	33.3333	-0.0666667	...
...	...	...	...

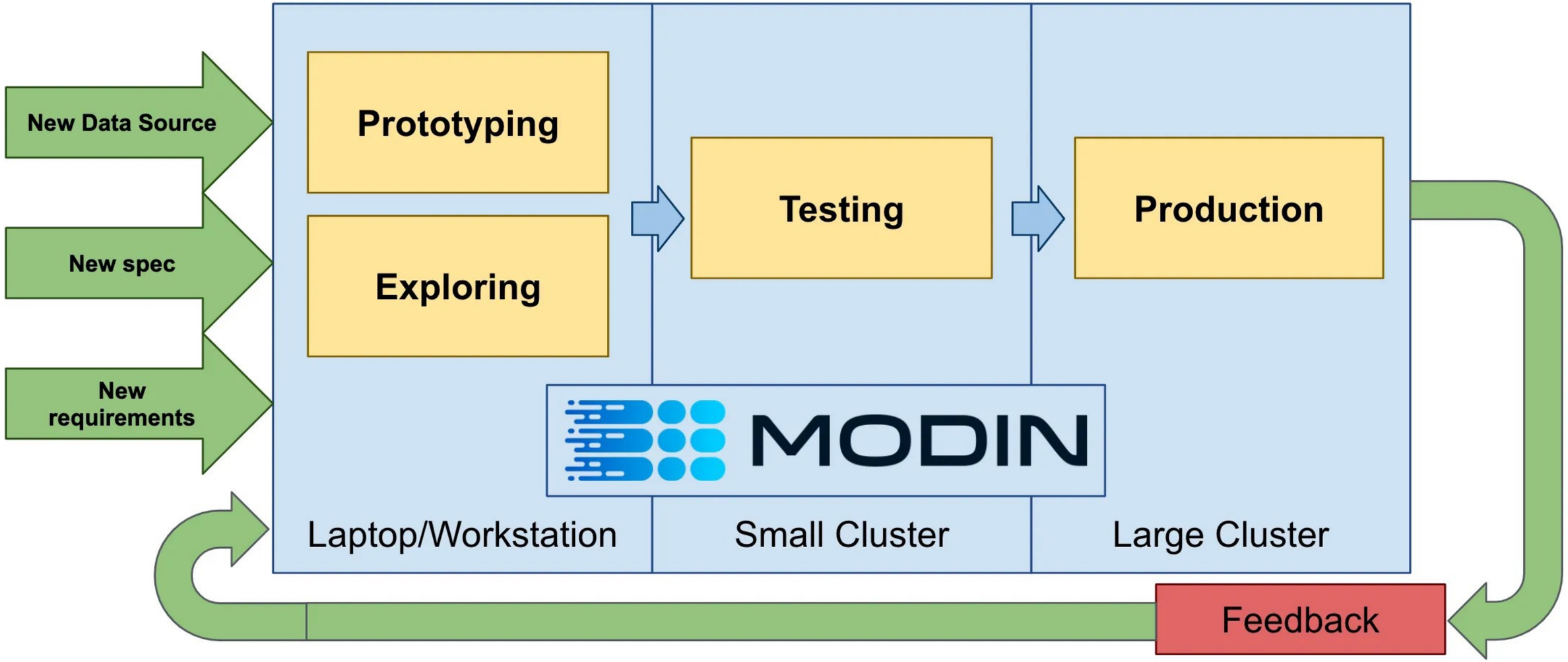
[D. Petersohn et al., 2020]

Problems Scaling: From Pandas to Other Solutions



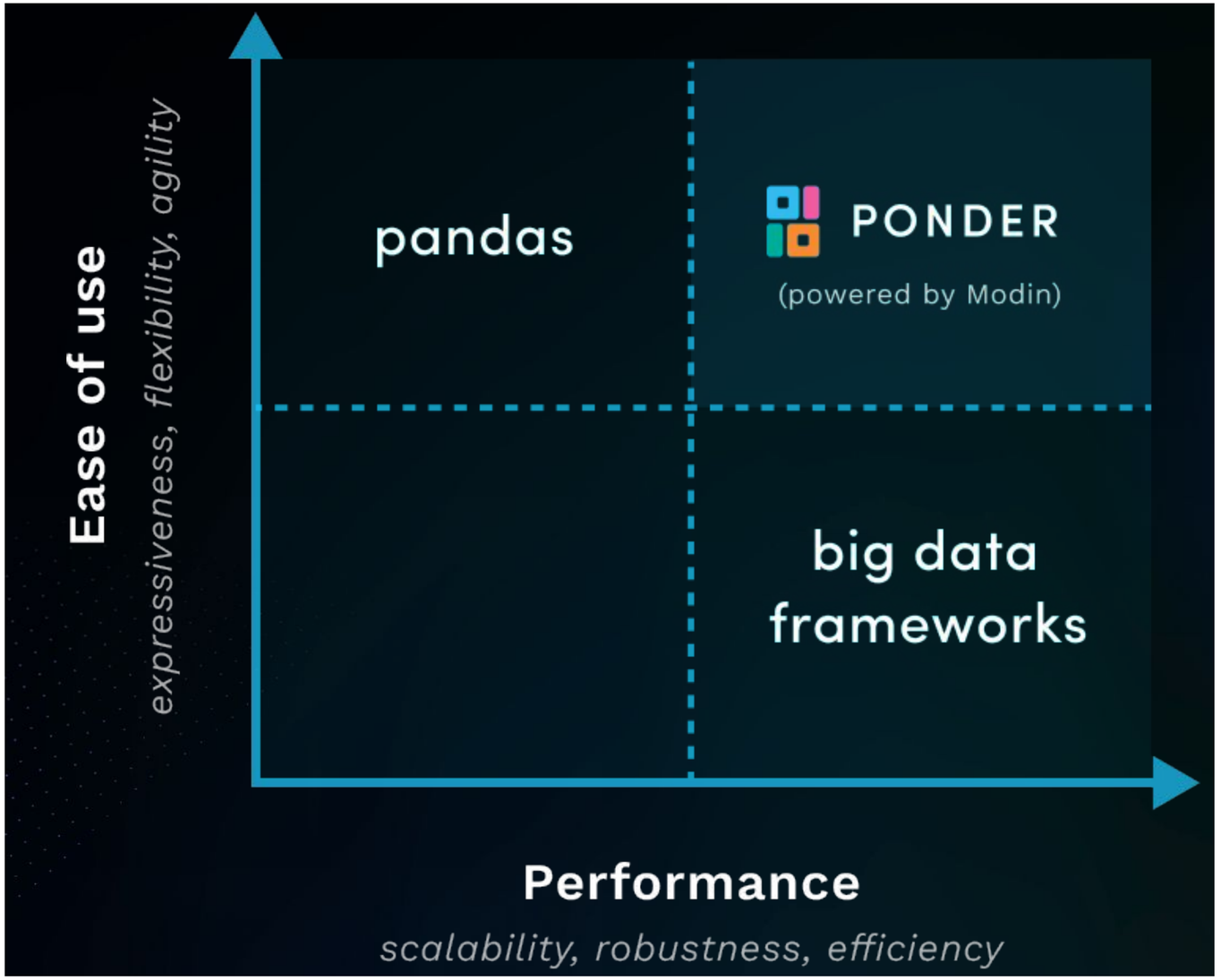
[D. Petersohn]

Modin as a Solution



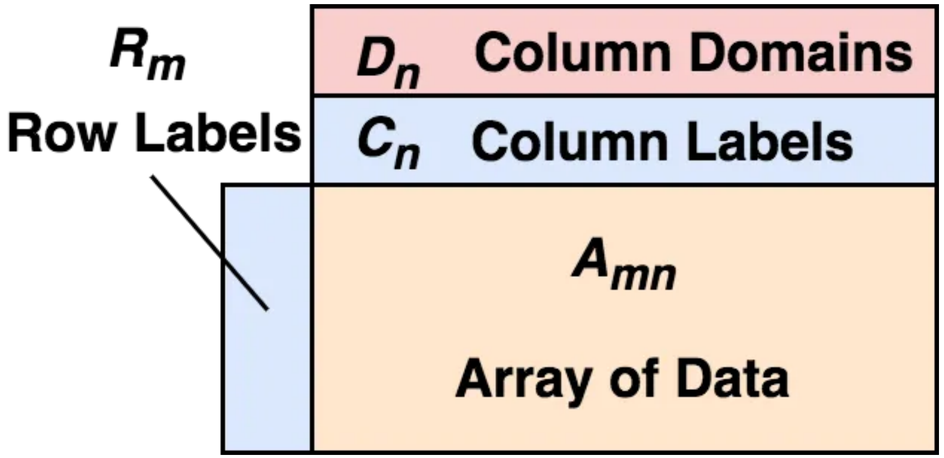
[D. Petersohn]

Modin Positioning



[D. Petersohn]

Dataframe Data Model



- Combines parts of matrices, databases, and spreadsheets
- Ordered, but not necessarily sorted
  - Rows and columns
- No predefined schema necessary
  - Types can be induced at runtime
- Typed Row/column labels
  - Labels can become data
- Indexing by label or row/column number
  - “Named notation” or “Positional notation”

[D. Petersohn]

Comparing Dataframes and Relational Stores

- Dataframe Characteristics
  - Ordered table
  - Named rows labels
  - A lazily-induced schema
  - Column names from $d \in Dom$
  - Column/row symmetry
  - Support for linear alg. operators
- Relational Characteristics
  - Unordered table
  - No naming of rows
  - Rigid schema
  - Column names from att
  - Columns and rows are distinct
  - No native support

[D. Petersohn et al., 2020]

Comparing Dataframes and Matrices

- Dataframe Characteristics
  - Heterogeneously typed
  - Numeric & non-numeric types
  - Explicit row and column labels
  - Support for rel. algebra operators
- Matrix Characteristics
  - Homogeneously typed
  - Only numeric types
  - No row or column labels
  - No native support

[D. Petersohn et al., 2020]

Dataframe Algebra

Operator	(Meta)data		Schema	Origin	Order	Description
SELECTION		×	static	REL	Parent	Eliminate rows
PROJECTION		×	static	REL	Parent	Eliminate columns
UNION		×	static	REL	Parent†	Set union of two dataframes
DIFFERENCE		×	static	REL	Parent†	Set difference of two dataframes
CROSS PRODUCT / JOIN		×	static	REL	Parent†	Combine two dataframes by element
DROP DUPLICATES		×	static	REL	Parent	Remove duplicate rows
GROUPBY		×	static	REL	New	Group identical values for a given (set of) attribute(s)
SORT		×	static	REL	New	Lexicographically order rows
RENAME	(×)		static	REL	Parent	Change the name of a column
WINDOW		×	static	SQL	Parent	Apply a function via a sliding-window (either direction)
TRANSPOSE	(×)	×	dynamic	DF	Parent◊	Swap data and metadata between rows and columns
MAP	(×)	×	dynamic	DF	Parent	Apply a function uniformly to every row
TOLABELS	(×)	×	dynamic	DF	Parent	Set a data column as the row labels column
FROMLABELS	(×)	×	dynamic	DF	Parent	Convert the row labels column into a data column

[D. Petersohn et al., 2020]

Pivot Example

Narrow Table (SALES)

Year	Month	Sales
2001	Jan	100
2001	Feb	110
2001	Mar	120
2002	Jan	150
2002	Feb	200
2002	Mar	250
2003	Jan	300
2003	Feb	310

Pivot $\longrightarrow$

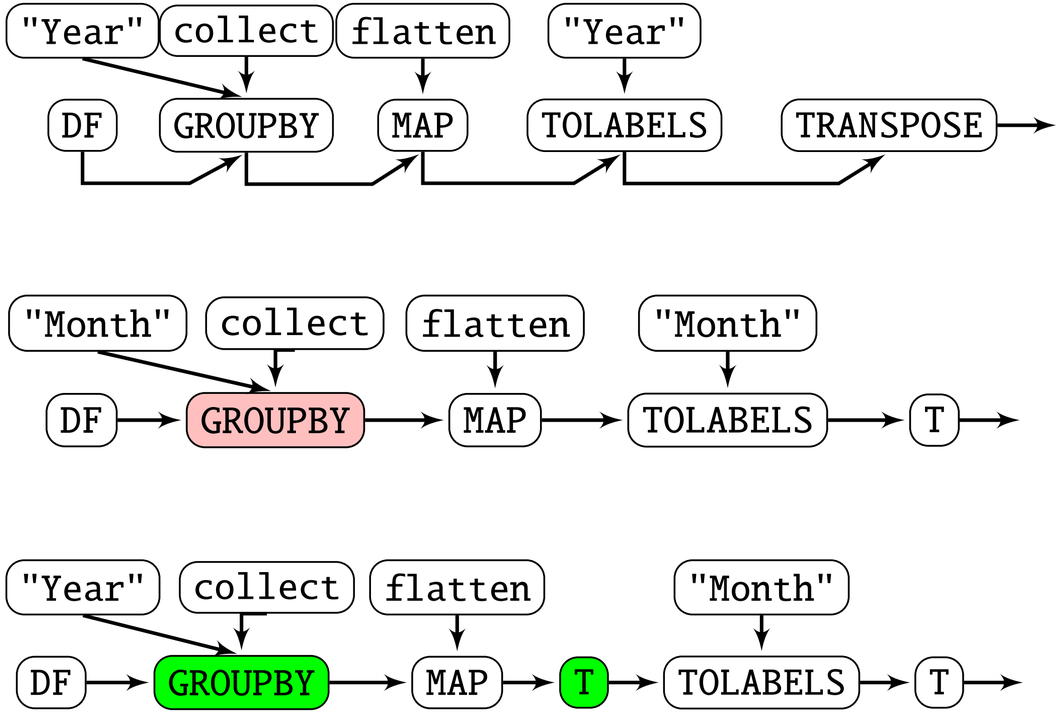
$\longleftarrow$ **Unpivot**

Wide Table of MONTHs

Month	2001	2002	2003
Jan	100	150	300
Feb	110	200	310
Mar	120	250	NULL

Wide Table of YEARS

Year	Jan	Feb	Mar
2001	100	110	120
2002	150	200	250
2003	300	310	NULL



[D. Petersohn et al., 2020]

Modin Challenges

- Massive API: 240+ operators, but with a lot of redundancy
- Parallel Execution: row-based, column-based, and block-based
- Data Model Challenges: Schema induction, reusing type info
- Order is important
- Supporting billions of columns: Row/Column equivalence (transpose)
- Metadata is data (and vice versa)
- Users want immediate feedback
- Users want to create queries incrementally

Assignment 4

- Work on Data Integration and Data Fusion
- Integrate artist datasets from different institutions (Met, NGA, AIC, CMA)
  - Integrate information based on ids and matching
- Record Matching:
  - Which artists are the same?
- Data Fusion:
  - Names
  - Dates
  - Nationalities

Test 2

- Upcoming… April 8
- Similar format, but more emphasis on topics we have covered including the research papers

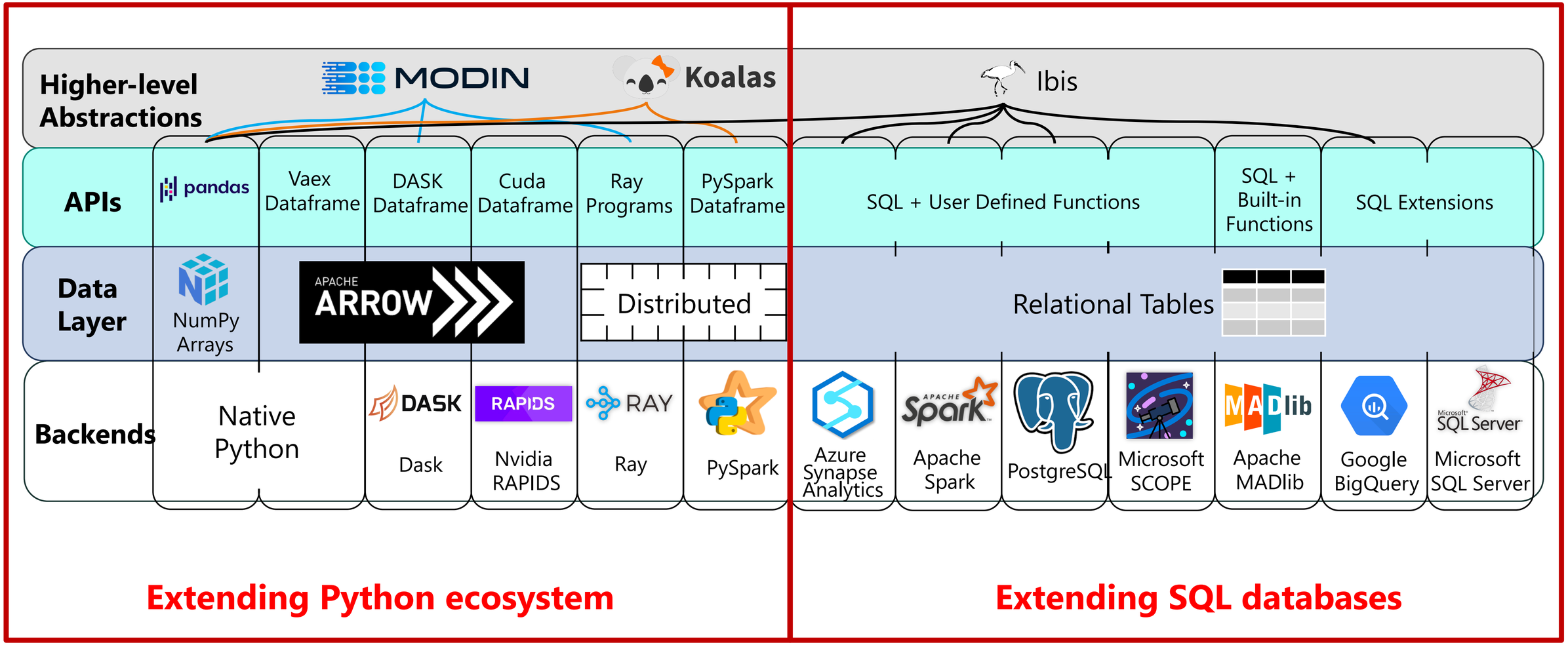
Dataframes, Databases, and the Cloud

- How do we take advantage of different architectures?
- Lots of work in scaling databases and specialized computational engines
- What is the code that people actually write?

Magpie: Python at Speed and Scale using Cloud Backends

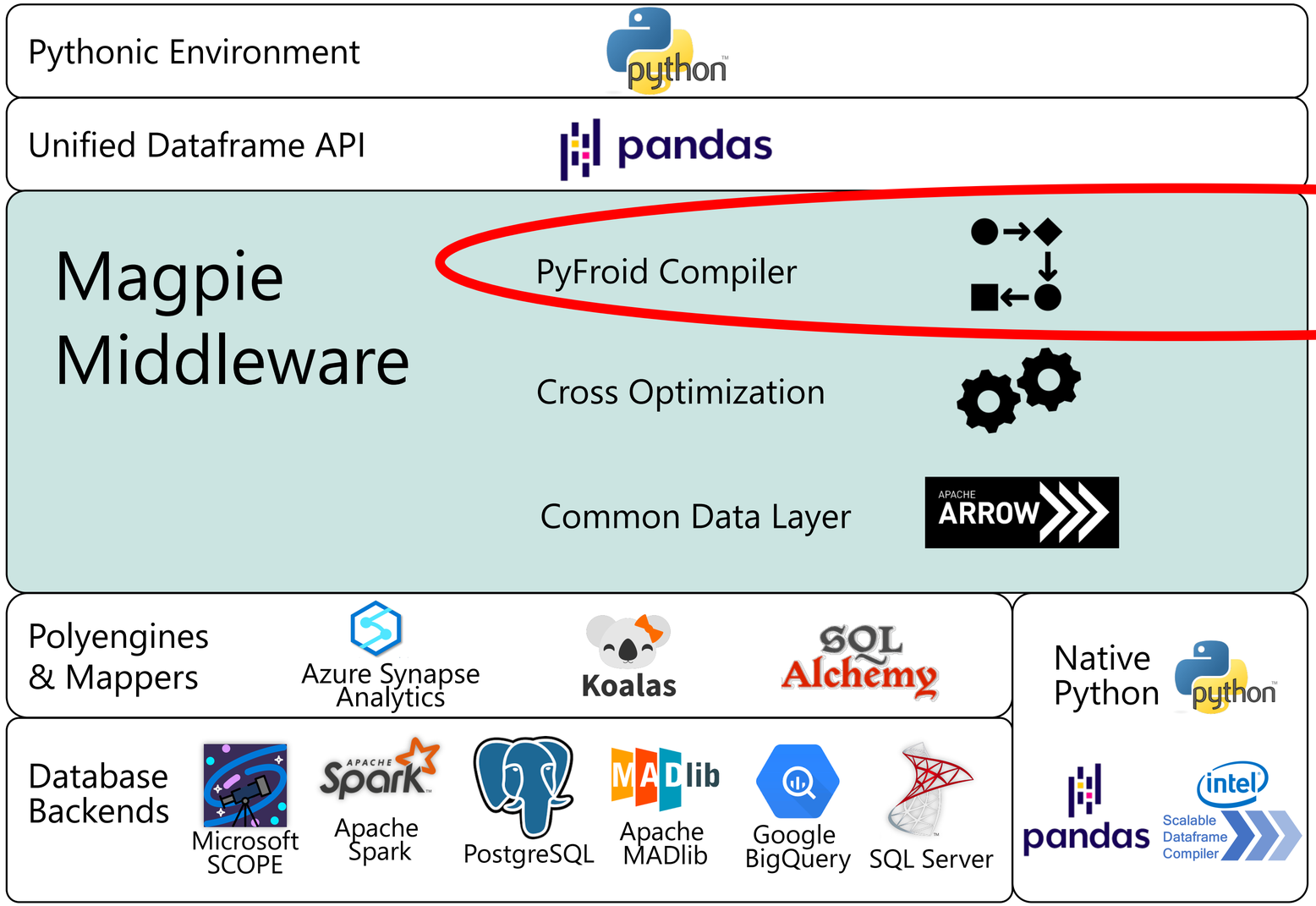
A. Jindal

Data Science Jungle



[A. Jindal et al., 2021]

Magpie Goals



Familiar Python surface

Ongoing standardization

Batching Pandas into large query expressions

Backend selection using past workloads

Cache commonly seen dataframes

Multi-backend environments and libraries

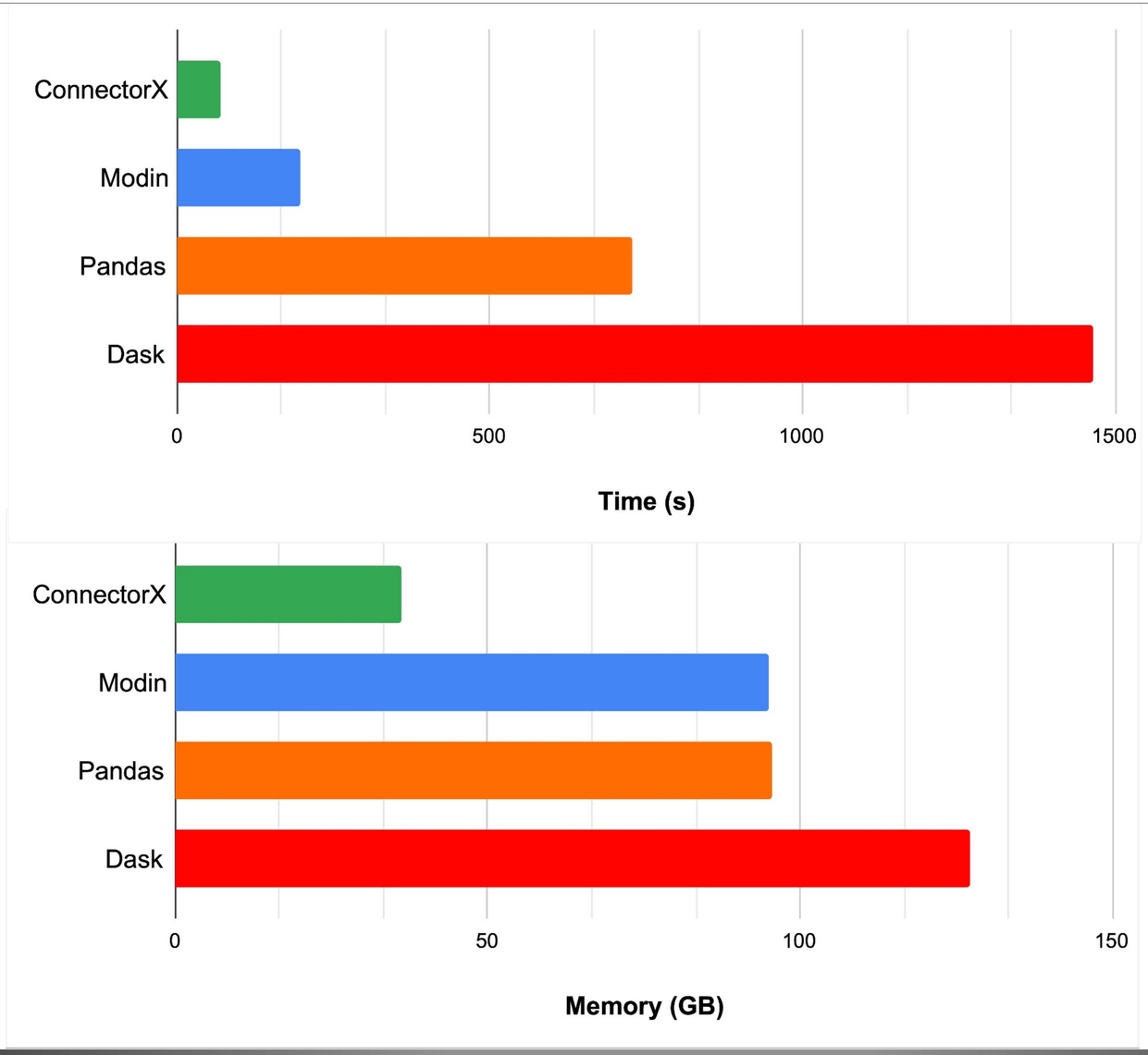
Cloud backends

ConnectorX: Databases to Dataframes

- Write read_sql queries but write SQL
- Written in Rust
- Returns a dataframe

```
query = f"""
SELECT l_orderkey,
 SUM(l_extendedprice * ( 1 – l_discount )) AS revenue,
 o_orderdate,
 o_shippriority
FROM customer,
 orders,
 lineitem
WHERE c_mktsegment = 'BUILDING'
 AND c_custkey = o_custkey
 AND l_orderkey = o_orderkey
 AND o_orderdate < DATE '1995-03-15'
 AND l_shipdate > DATE '1995-03-15'
GROUP BY l_orderkey,
 o_orderdate,
 o_shippriority
"""
df = read_sql("postgresql://postgres:postgres@localhost:5432/tpch", query,
              partition_on="l_orderkey", partition_num=4)
```

ConnectorX Speed & Memory



[X. Wang, 2022]

Improvements in ConnectorX

- Written in native language (Rust)
- Copy exactly once (even during parallel computations)
- CPU cache-friendly: process in a streaming fashion

An Opinionated Introduction to Polars

Nico Kreiling

Handling Large Data with Polars

Etienne Bacher

Discussion

- Data in the cloud and local exploration
- Languages: SQL or Pandas or Ibis or….?