Advanced Data Management (CSCI 490/680)

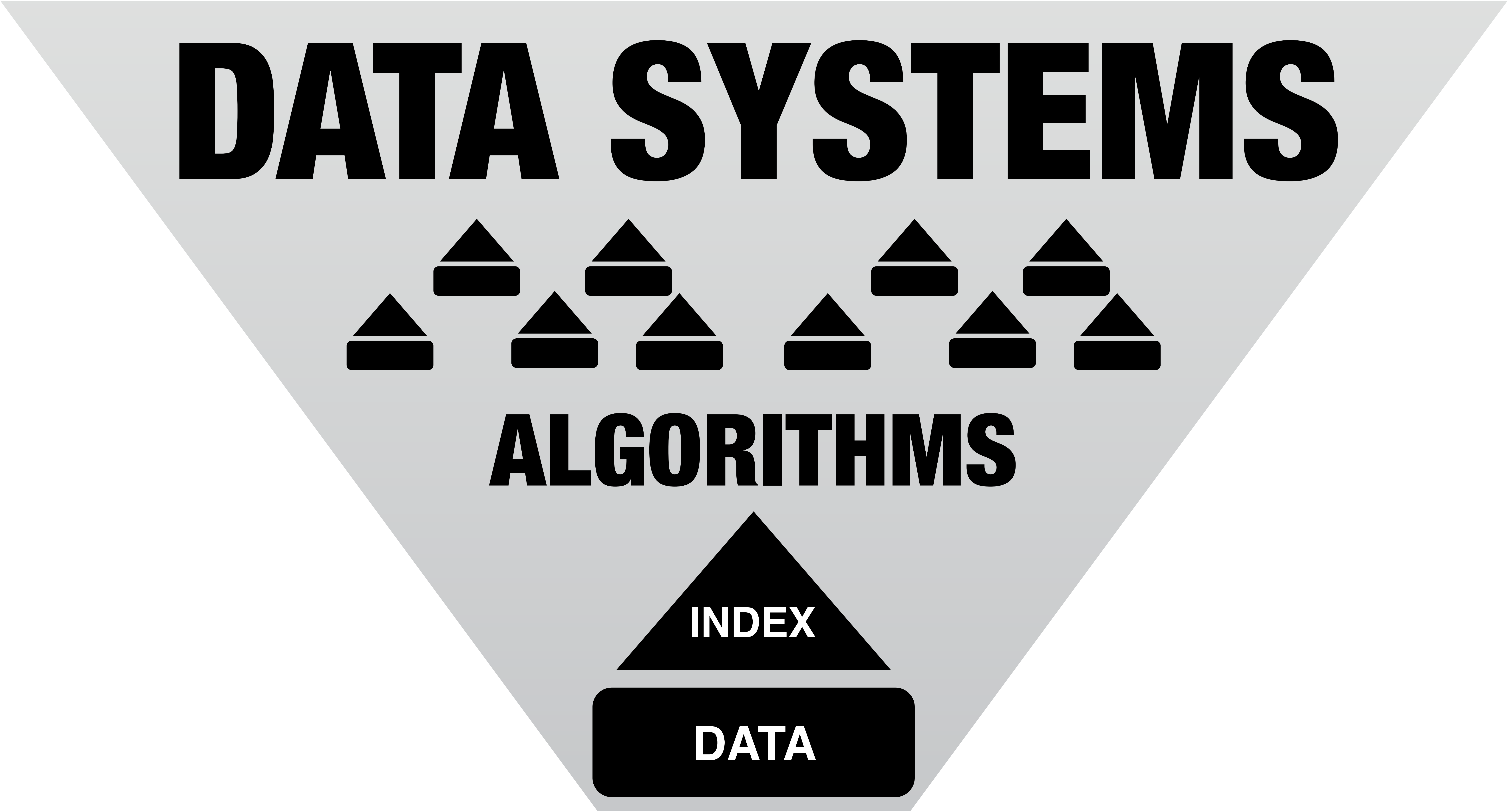
Review

Dr. David Koop

Final Exam

- Tuesday, May 5 from 4-5:50pm
- Online
- Similar format to Test 2
- Comprehensive but with more focus on last few weeks of class
- Contact me with questions:
  - Email
  - Setup a time to talk via Blackboard

Data systems rely on algorithms



DATA SYSTEMS

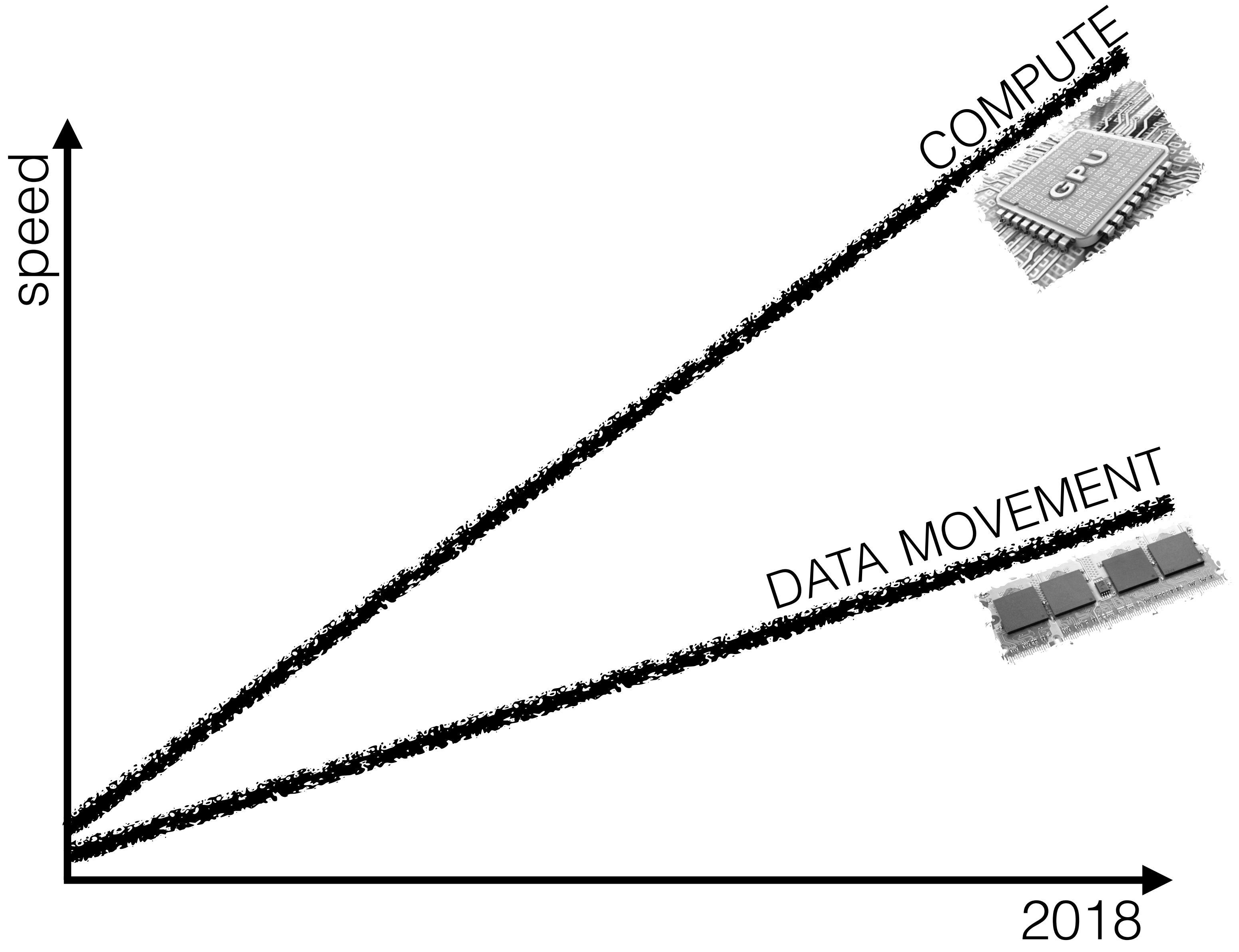
ALGORITHMS

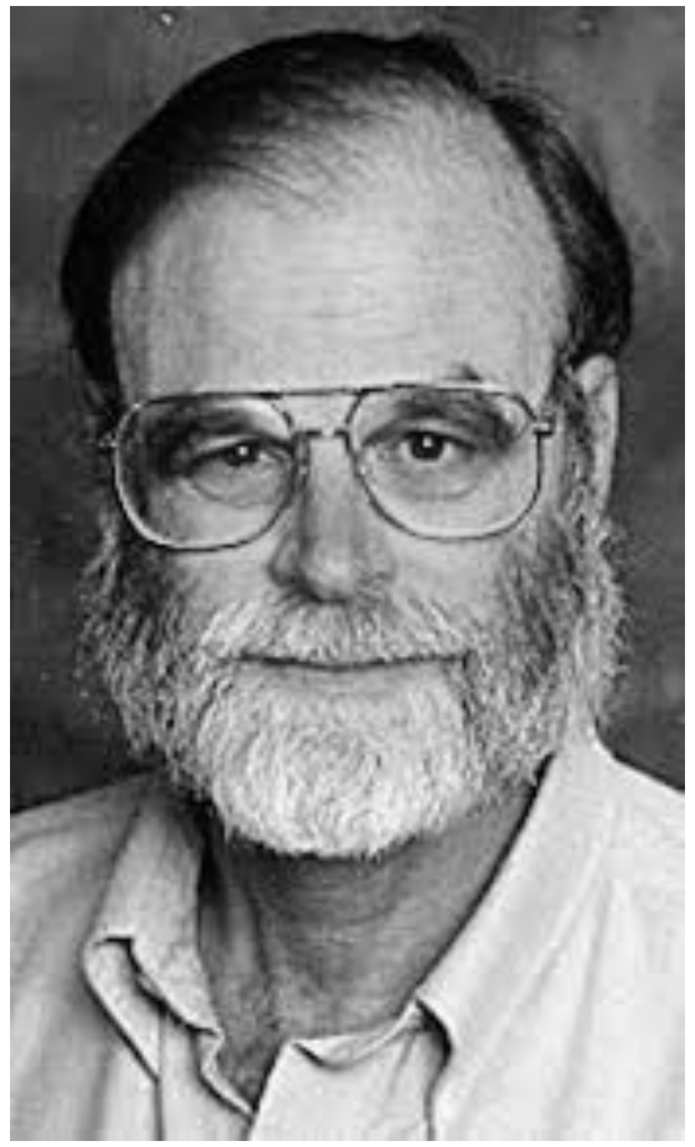
INDEX

DATA

[S. Idreos, 2019]

Data structures define performance



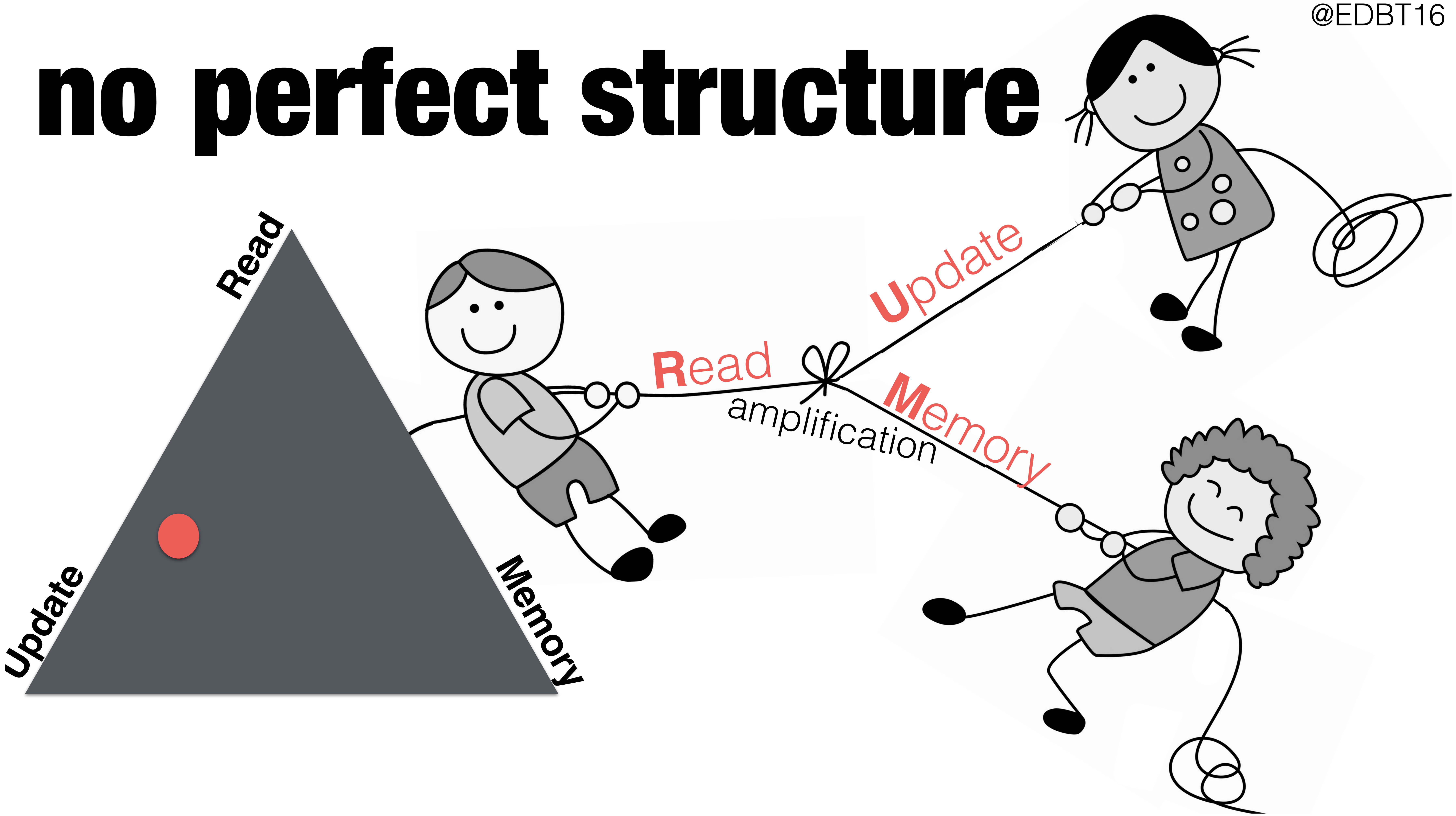


register = this room
caches = this city
memory = nearby city
disk = Pluto

Jim Gray, Turing Award 1998

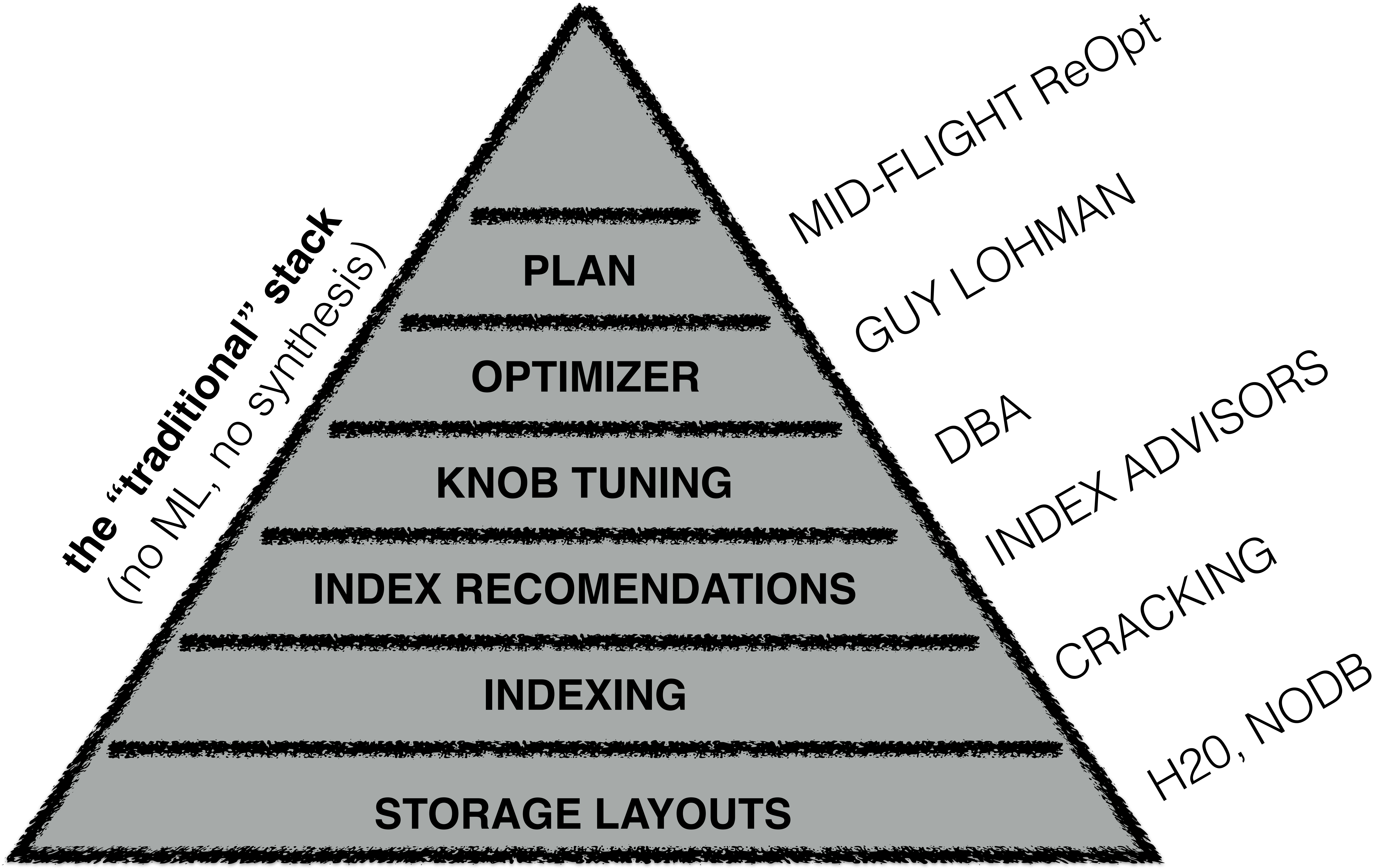
[S. Idreos, 2019]

Tradeoffs in each structure



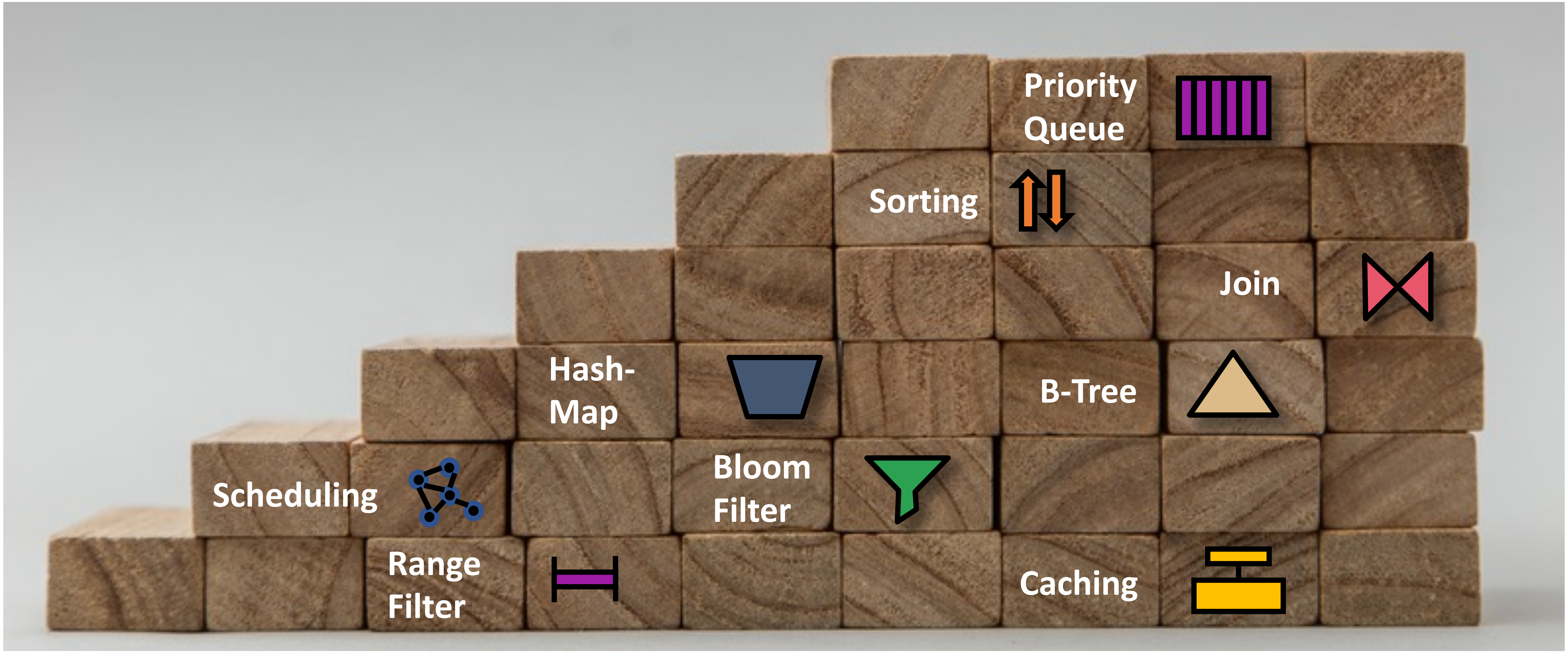
[S. Idreos, 2019]

"Traditional" Database Research

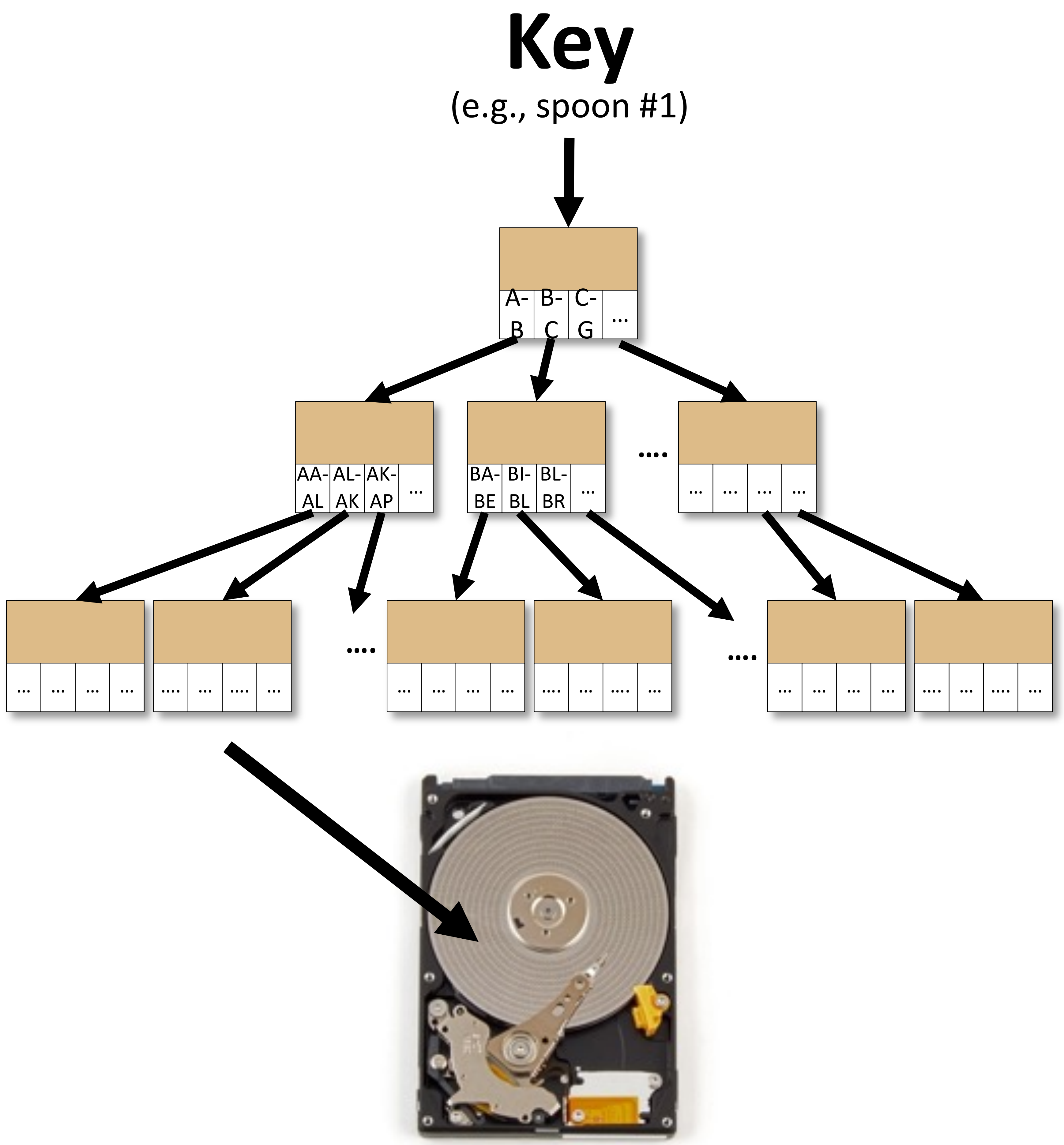


[S. Idreos, 2019]

Learned Data Structures and Algorithms

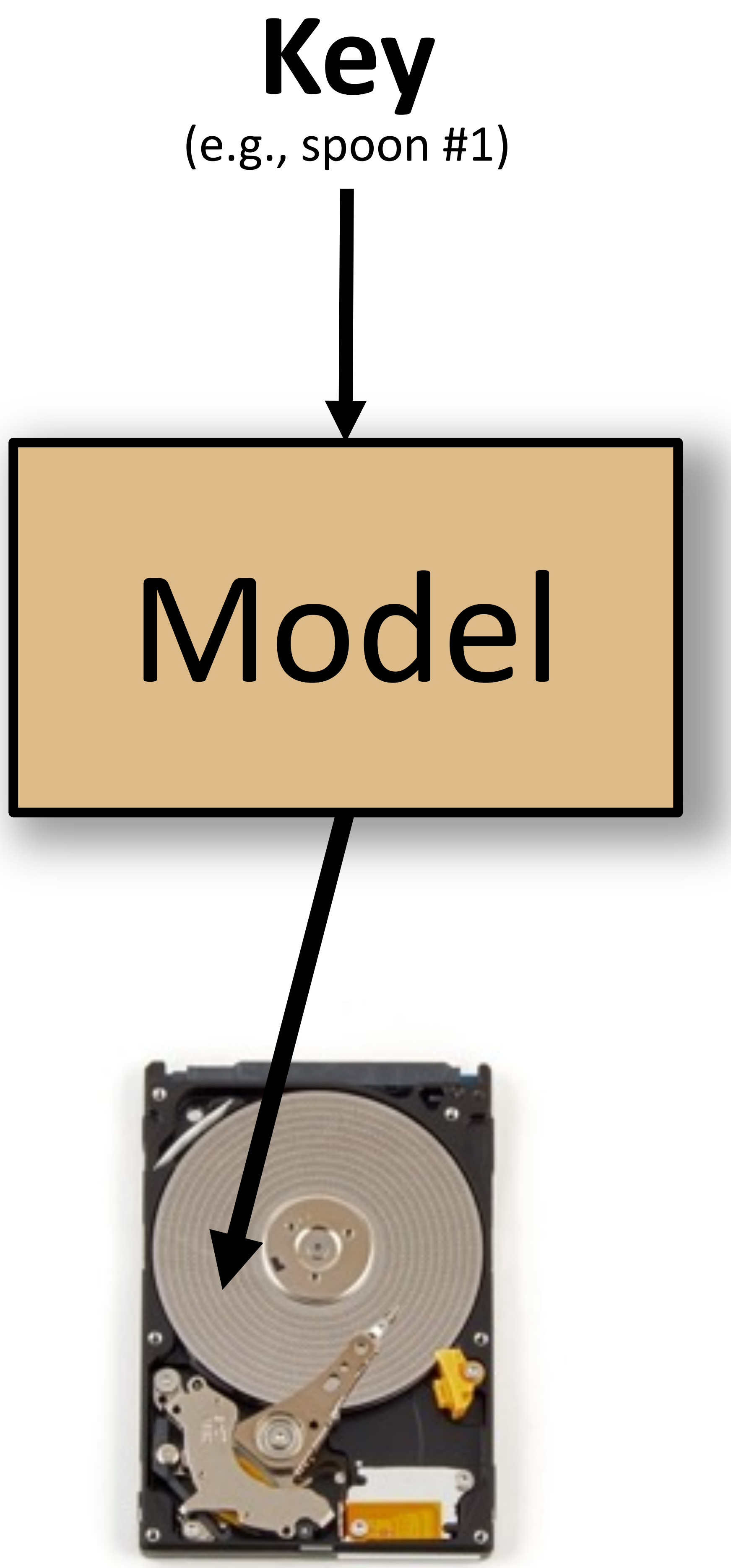


B-Tree



Key

(e.g., spoon #1)



Key

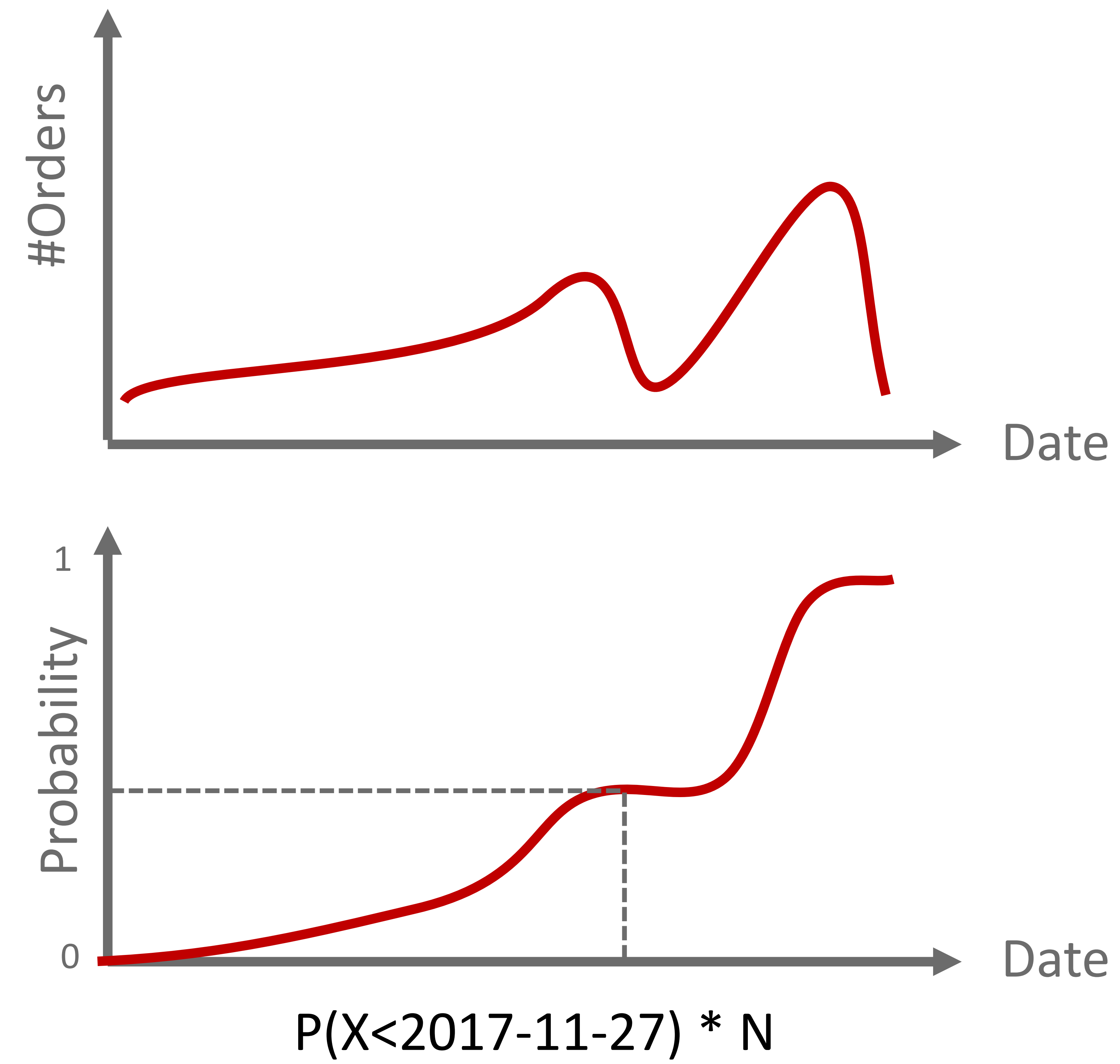
(e.g., spoon #1)

[T. Kraska, 2019]

Model to Predict Data's Location on Disk

Frequency Distribution

Cumulative Distribution Function (CDF)

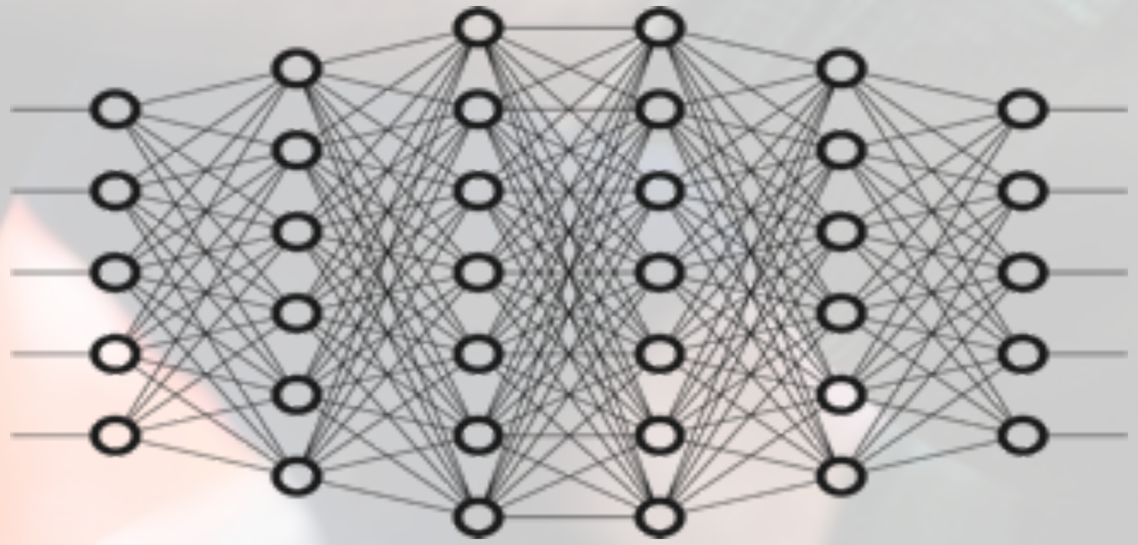


date
2017-01-01
2017-01-02
2017-01-02
2017-01-03
2017-01-03
2017-01-04
2017-01-04
2017-01-05
2017-01-05
2017-01-06
2017-01-07
2017-01-09
2017-01-09
2017-01-09
2017-01-10
2017-01-10
2017-01-11
2017-01-12
2017-01-13
2017-01-14
2017-01-15
2017-01-16
2017-01-17
2017-01-18
2017-01-19
2017-01-20
2017-01-21
2017-01-22
2017-01-22
2017-01-22
2017-01-23
2017-01-24
2017-01-24
2017-01-26
2017-01-26
2017-01-26
2017-01-28
2017-01-29
2017-01-30
2017-01-30
2017-01-30
2017-01-31
2017-01-31
2017-02-01
2017-02-01
2017-02-02
2017-02-04
2017-02-05
2017-02-05
2017-02-06
2017-02-06
2017-02-06
2017-02-07
2017-02-07
2017-02-08
2017-02-08
2017-02-08
2017-02-09
…
2017-11-27
2017-11-27
2017-11-27
2017-11-28
2017-11-28
…

[T. Kraska, 2019]

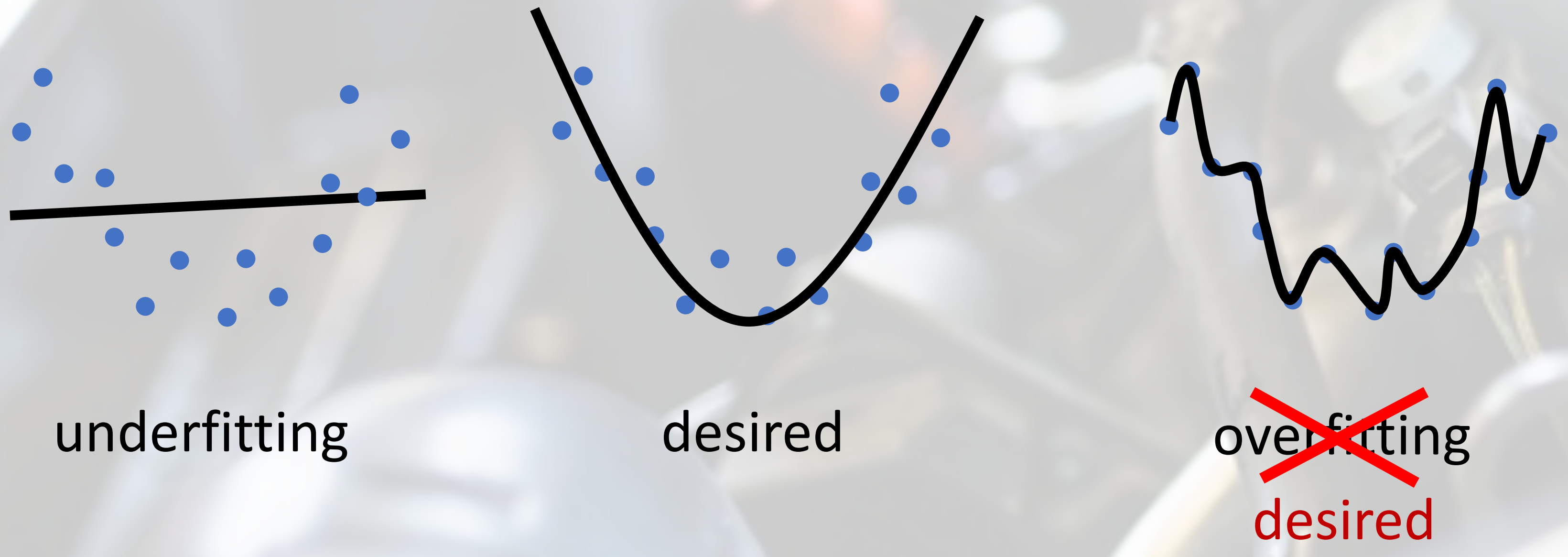
Challenges

Traditional model architectures do not work

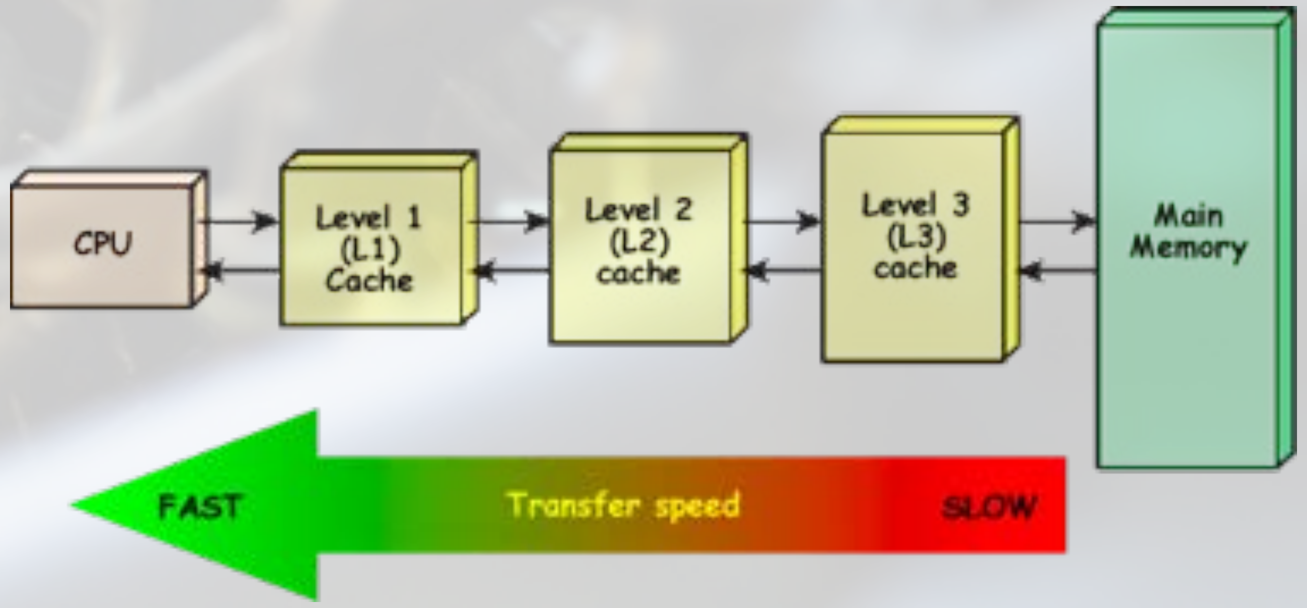


Frameworks are not designed for nano-second execution

Overfitting can be good

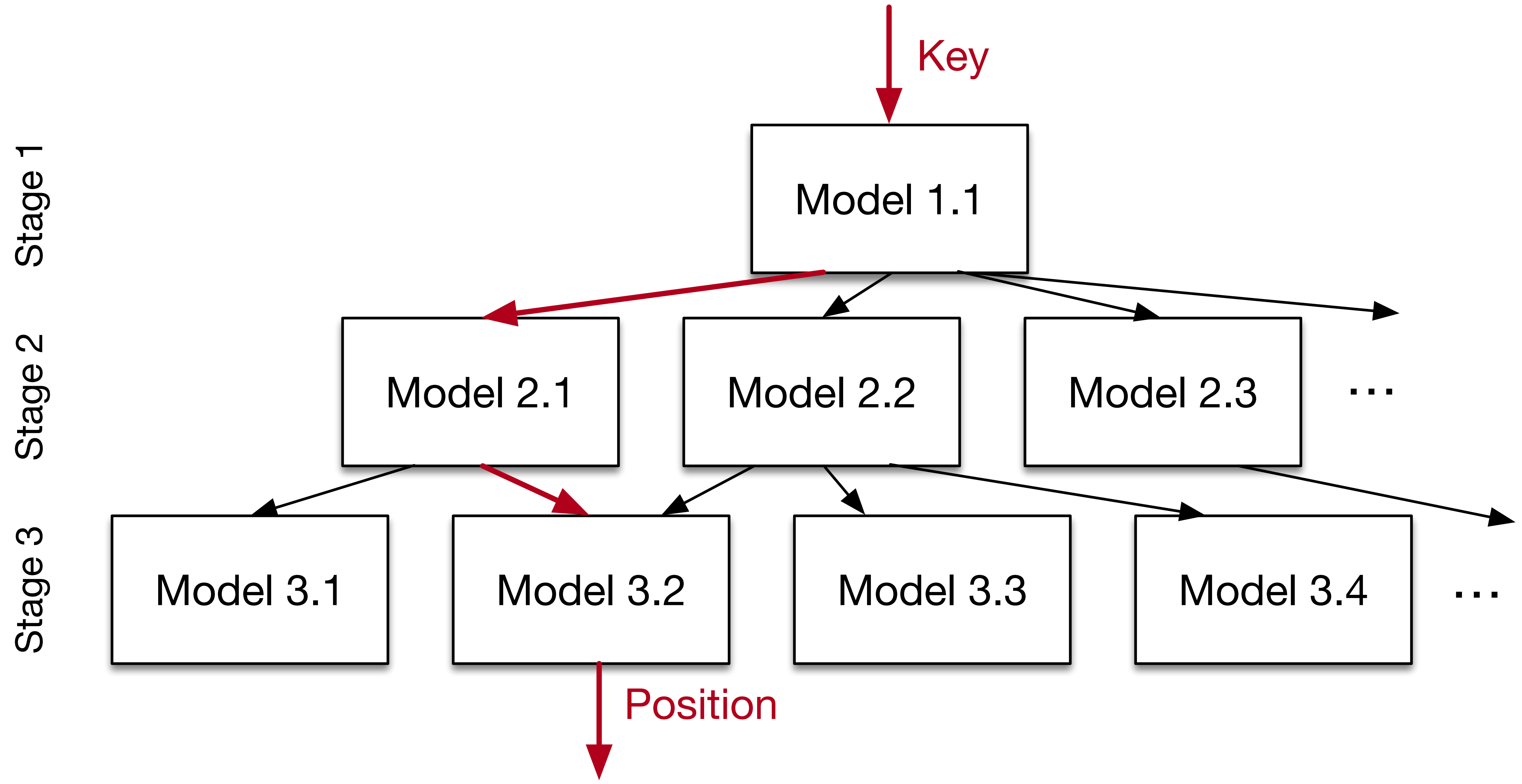


ML+System Co-Design



[T. Kraska, 2019]

Recursive Model Index (RMI)

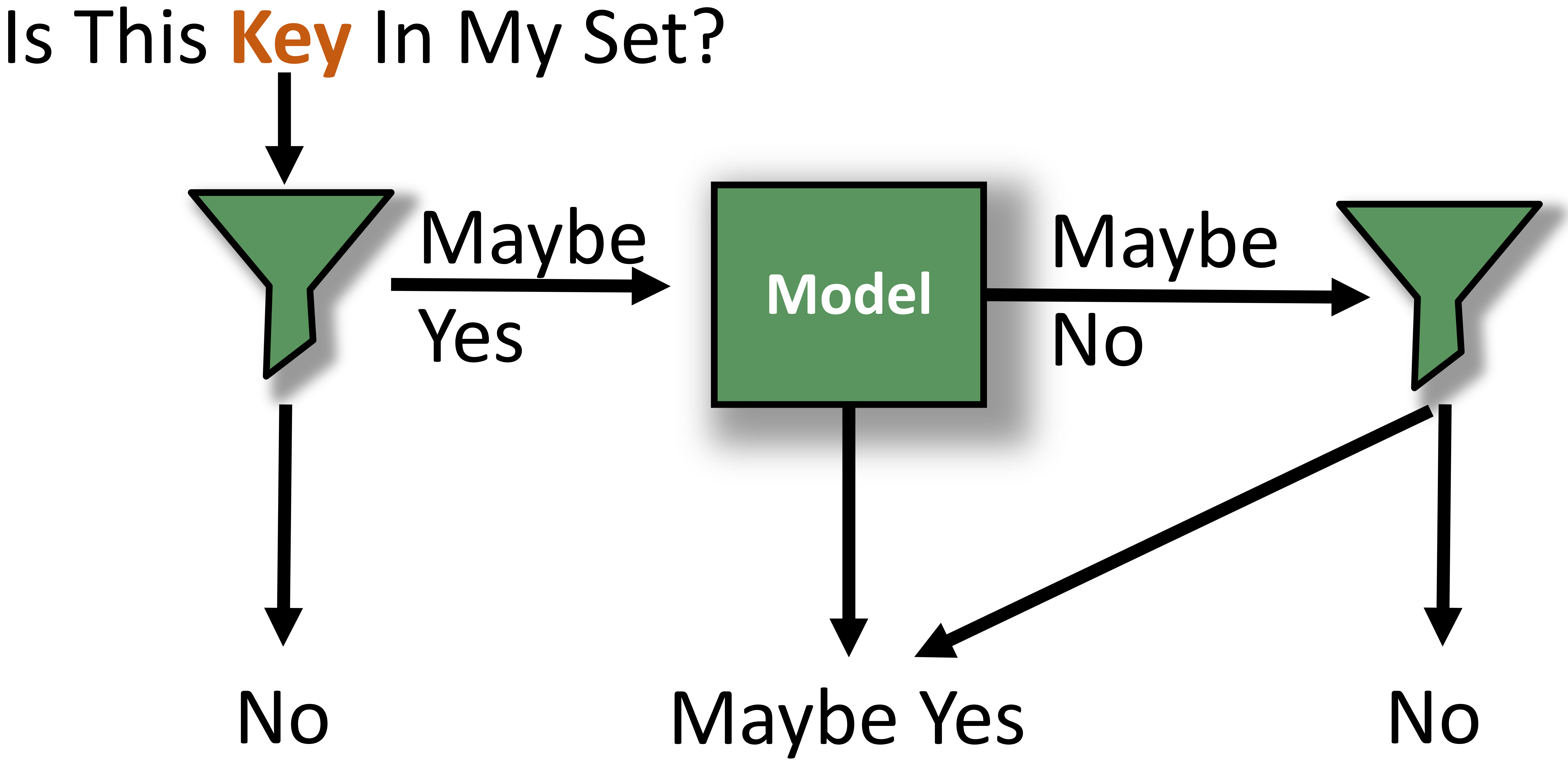


2-Stage RMI with Linear Model

```
pos_0 = a_0 + b_0 * key
pos_1 = m_1[pos_0].a + m_1[pos_0].b * key
record = local-search(key, pos_1)
```

[T. Kraska, 2019]

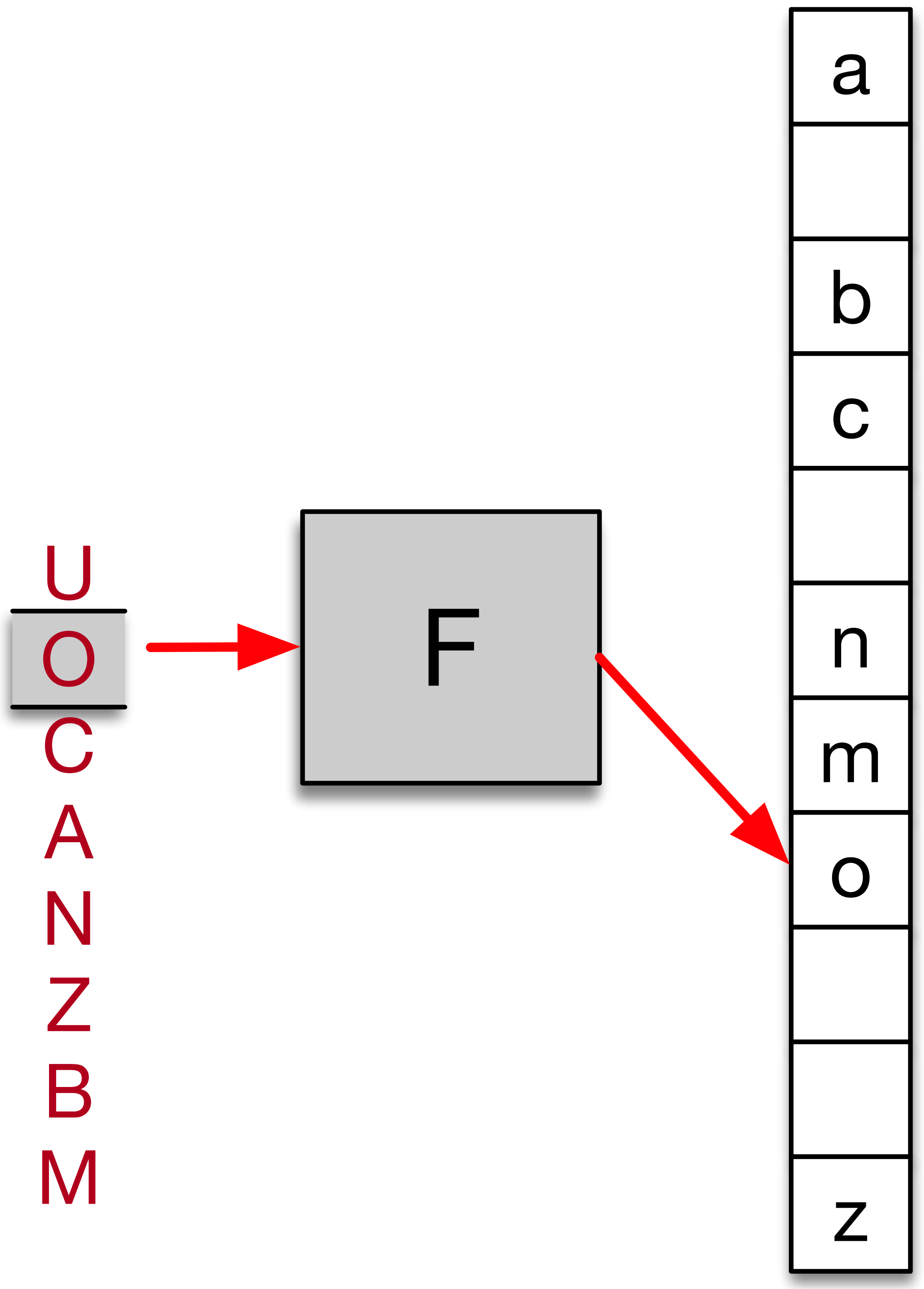
Sandwiched Bloom Filter



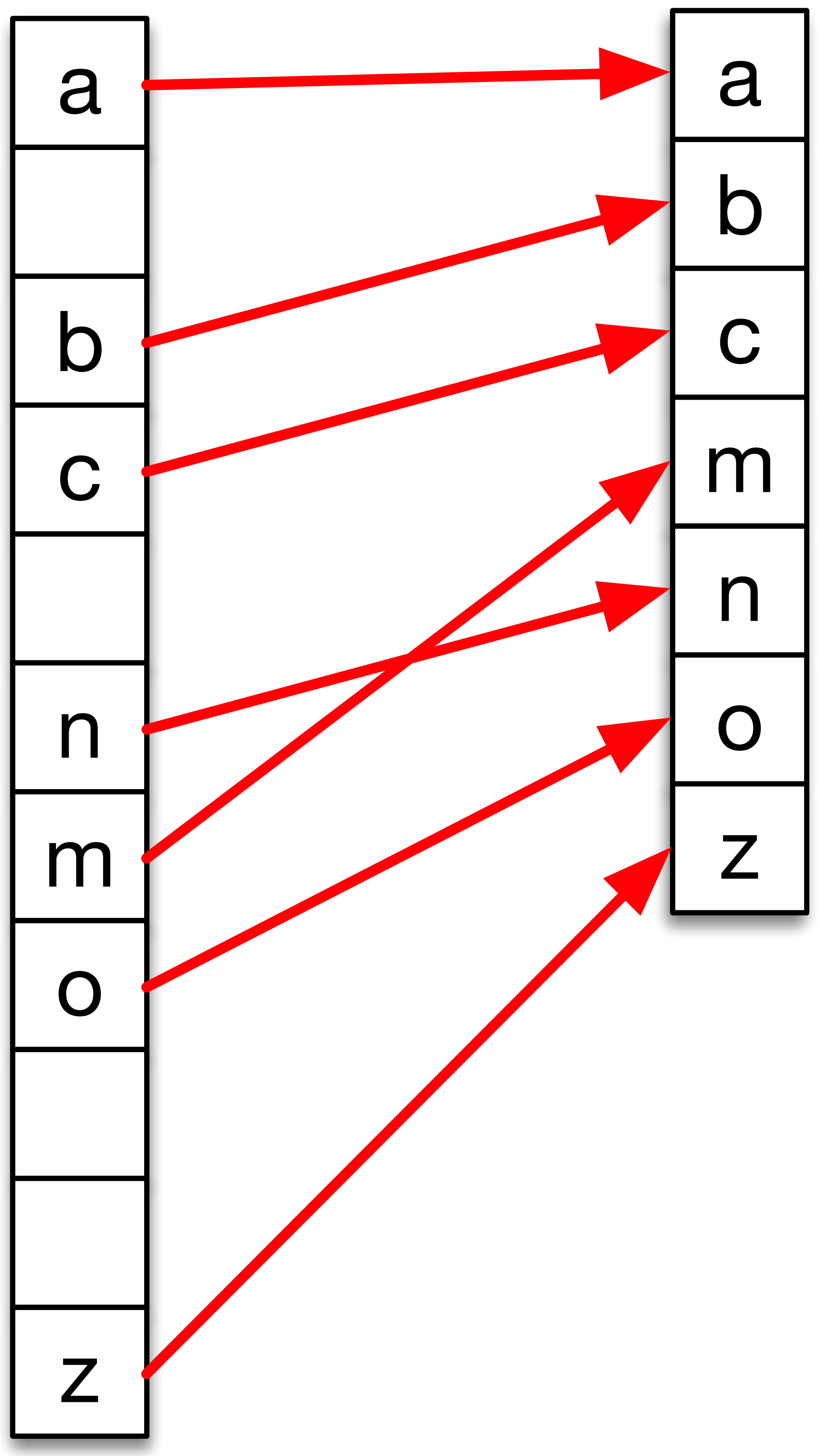
[M. Mitzenmacher, 2018 via T. Kraska, 2019]

Sorting

(a) CDF Model Pre-Sorts



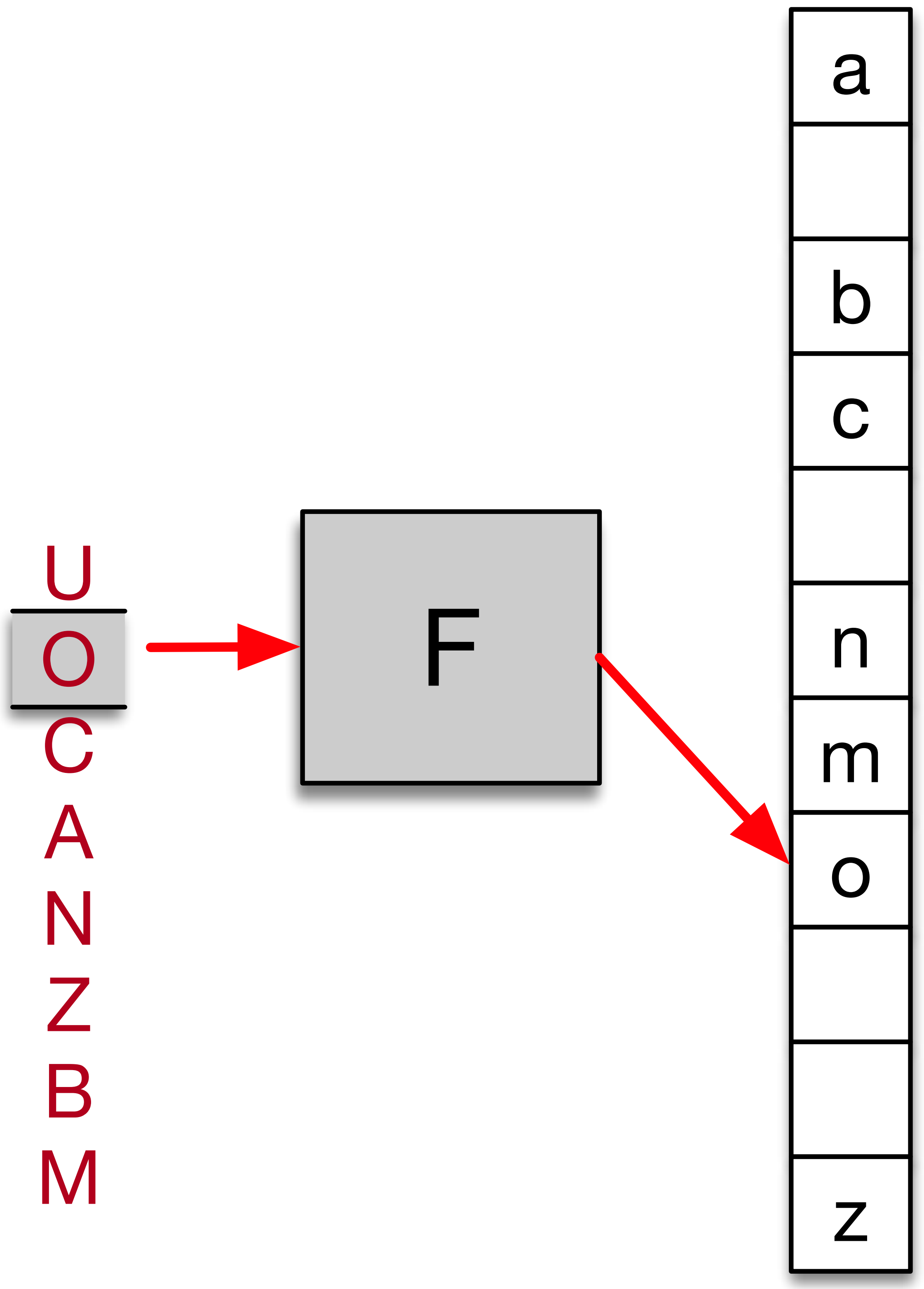
(b) Compact & local sort

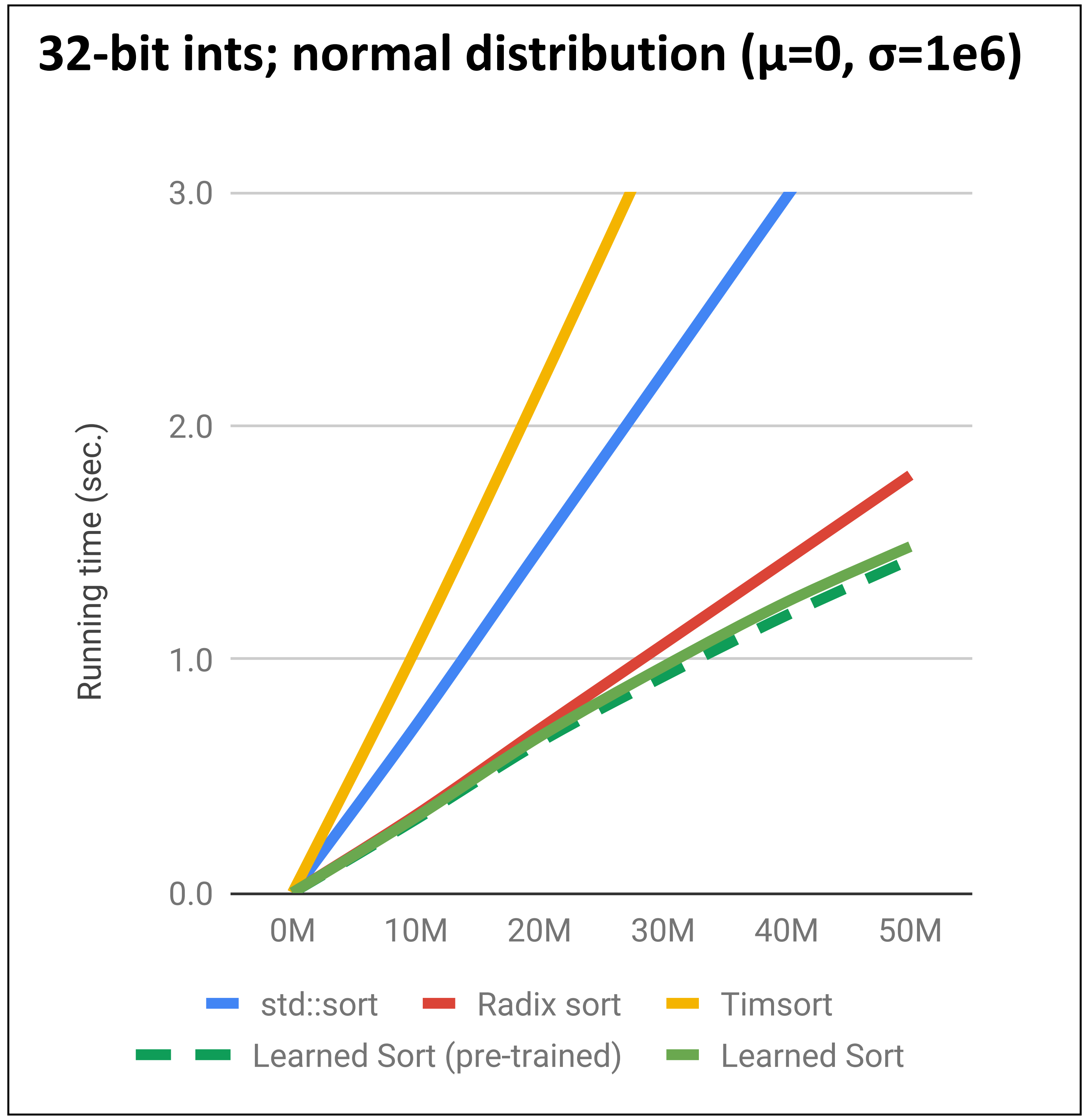


[T. Kraska, 2019]

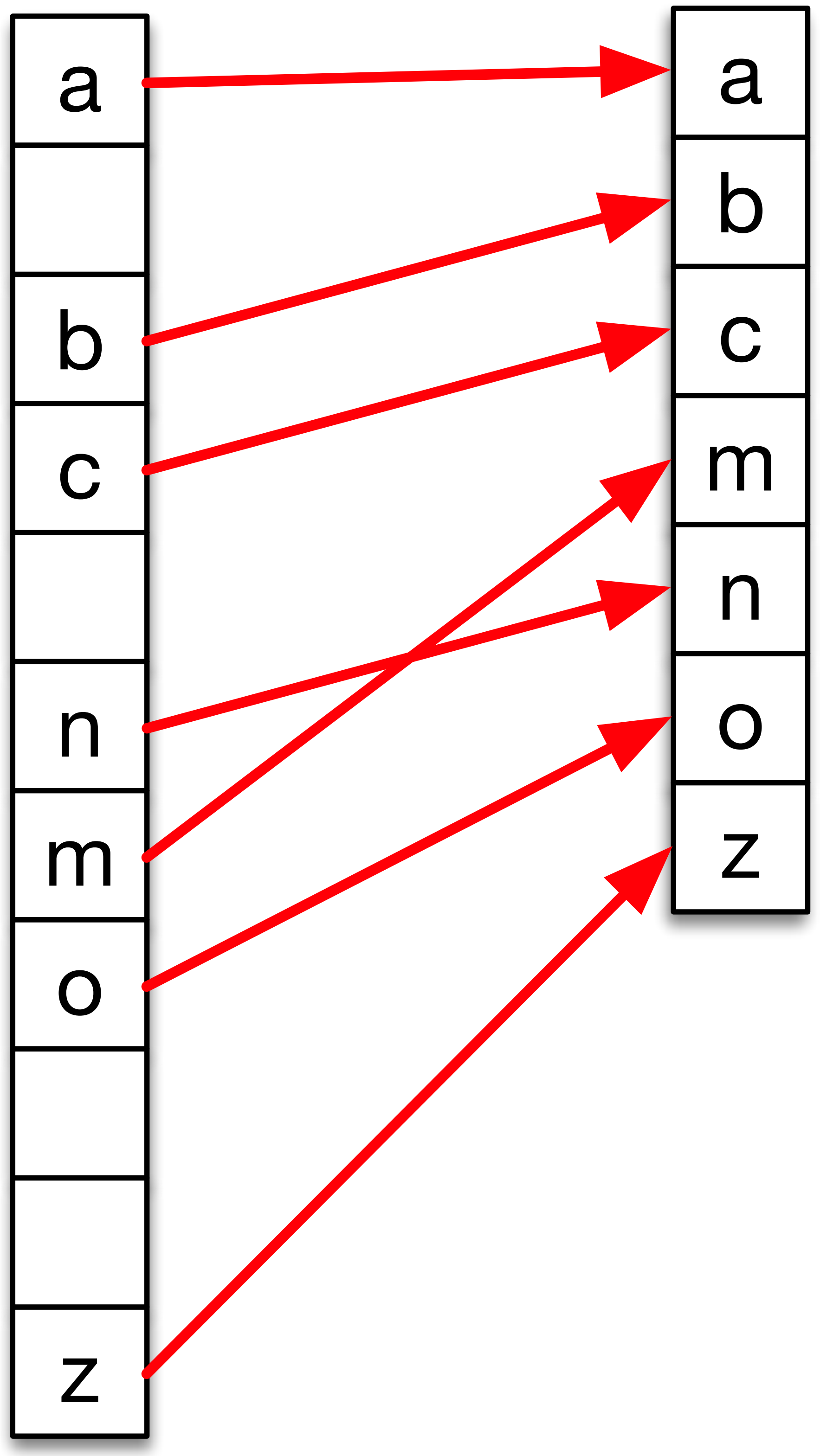
Sorting

(a) CDF Model Pre-Sorts



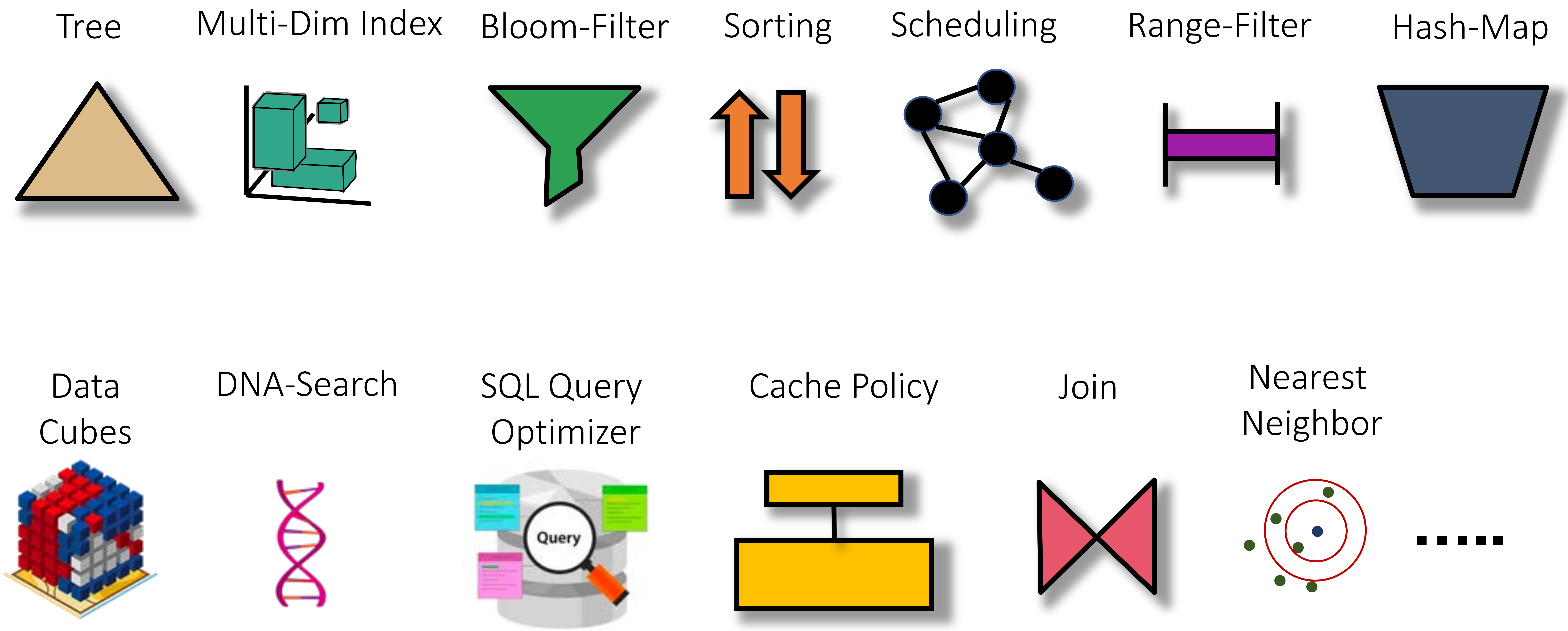


(b) Compact & local sort



[T. Kraska, 2019]

More…



Tree

Multi-Dim Index

Bloom-Filter

Sorting

Scheduling

Range-Filter

Hash-Map

Data Cubes

DNA-Search

SQL Query Optimizer

Cache Policy

Join

Nearest Neighbor

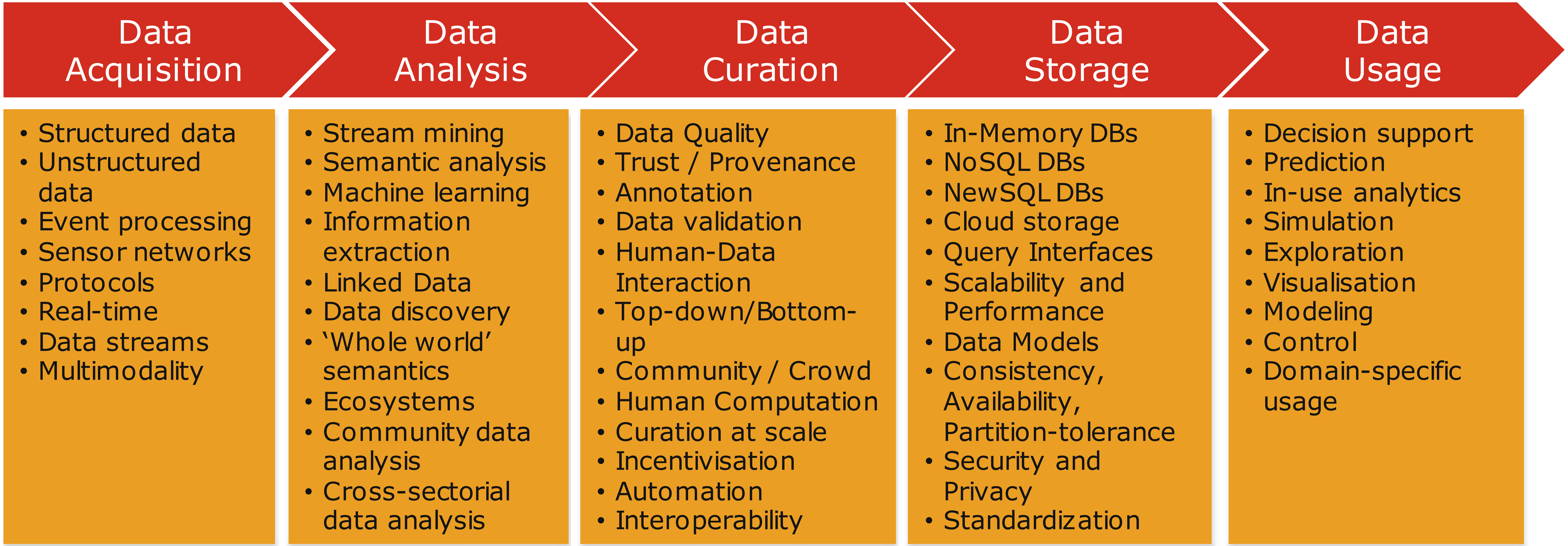
[T. Kraska, 2019]

Review

Re

What did we do this semester?

What's involved in dealing with data?



Data Acquisition

- Structured data
- Unstructured data
- Event processing
- Sensor networks
- Protocols
- Real-time
- Data streams
- Multimodality

Data Analysis

- Stream mining
- Semantic analysis
- Machine learning
- Information extraction
- Linked Data
- Data discovery
- ‘Whole world’ semantics
- Ecosystems
- Community data analysis
- Cross-sectorial data analysis

Data Curation

- Data Quality
- Trust / Provenance
- Annotation
- Data validation
- Human-Data Interaction
- Top-down/Bottom-up
- Community / Crowd
- Human Computation
- Curation at scale
- Incentivisation
- Automation
- Interoperability

Data Storage

- In-Memory DBs
- NoSQL DBs
- NewSQL DBs
- Cloud storage
- Query Interfaces
- Scalability and Performance
- Data Models
- Consistency, Availability, Partition-tolerance
- Security and Privacy
- Standardization

Data Usage

- Decision support
- Prediction
- In-use analytics
- Simulation
- Exploration
- Visualisation
- Modeling
- Control
- Domain-specific usage

[Big Data Value Chain, Curry et al., 2014]

Python!

- Just assign expressions to variables, no typing

```
a = 12
a = "abc"
b = a + "de"
```

- Functions defined using def, called using parenthesis:

```
def hello(name1="Joe", name2="Jane"):
    print(f"Hello {name1} and {name2}")
hello(name2="Mary")
```

- Always indent blocks (if-else-elif, while, for, etc.):

```
 z = 20
if x > 0:
    if y > 0:
        z = 100
else:
    z = 10
```

Python Containers

- List: `[1,"abc",12.34]`
- Tuple: `(1, "abc", 12.34)`
- Indexing/Slicing:
    - `x[0], x[:-1], x[1:2], x[::2]`
- Set: `{1, "abc", 12.34}`
- Dictionary: `{'x': 1, 'y': "abc", 'z': 12.34}`
- Mutable vs. Immutable
- Stored by reference
- Iterators: objects that traverse containers, just know how to get next element
- You cannot index/slice an iterator (`d.values()[-1]` doesn't work)

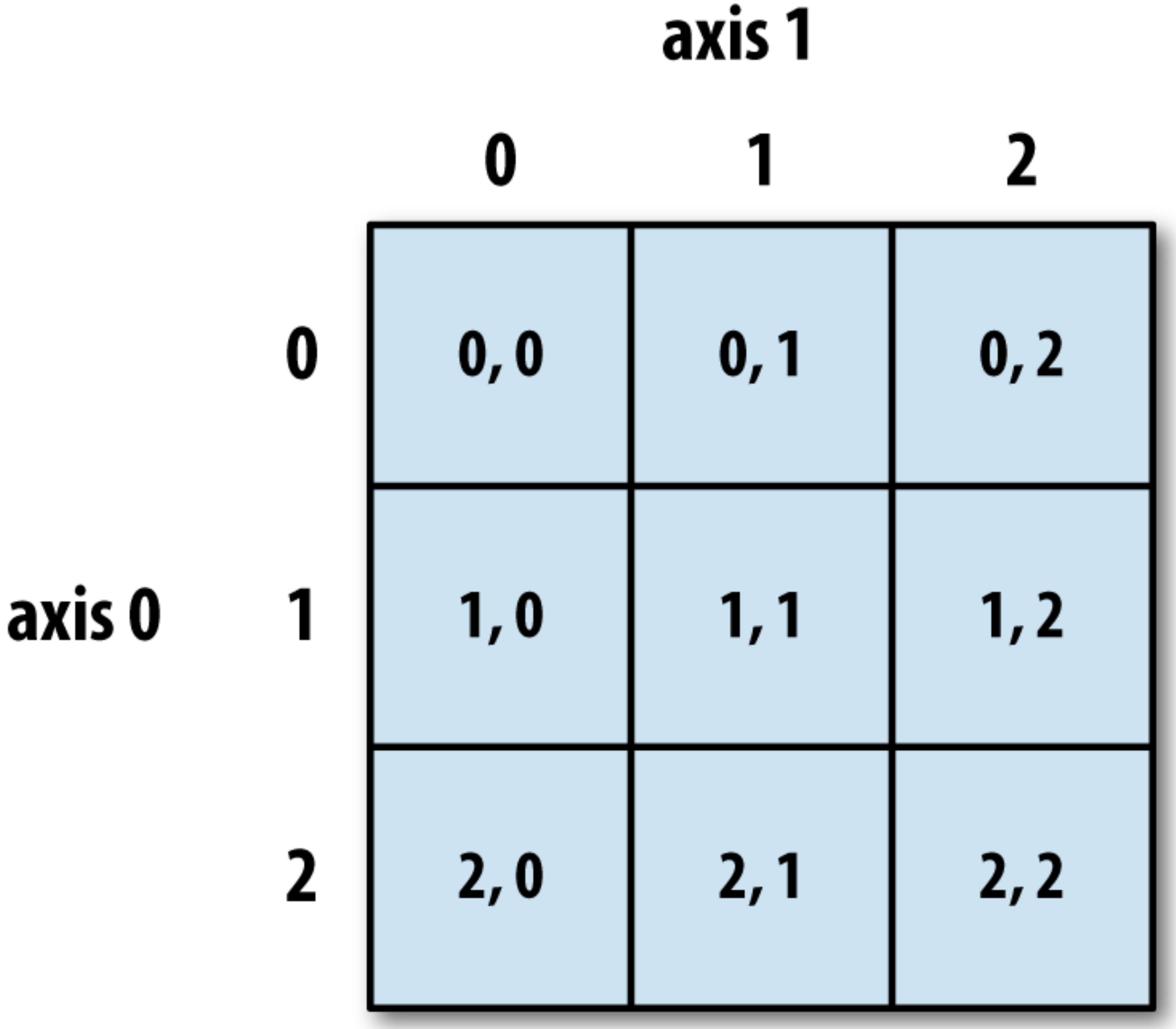
Comprehensions

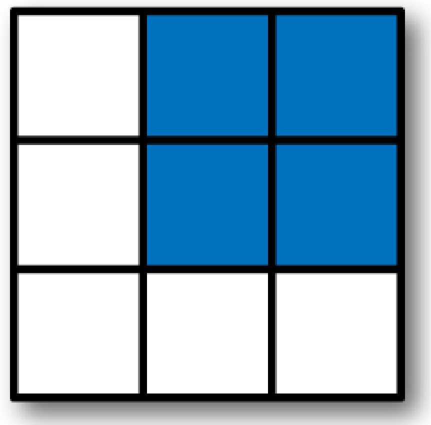
- List Comprehensions:
  - `squares = [i**2 for i in range(10)]`
- Dictionary Comprehensions:
  - `squares = {i: i**2 for i in range(10)}`
- Set Comprehensions:
  - `squares = {i**2 for i in range(10)}`
- Comprehensions allow filters:
  - `squares = [i**2 for i in range(10) if i % 2 == 0]`

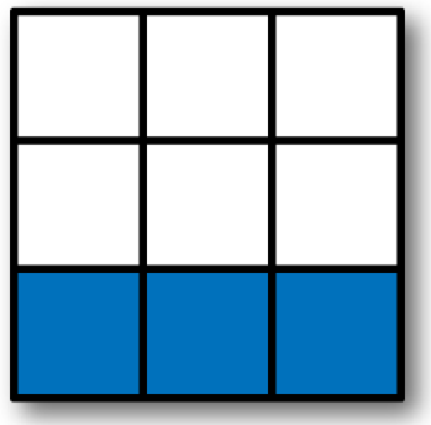
JupyterLab

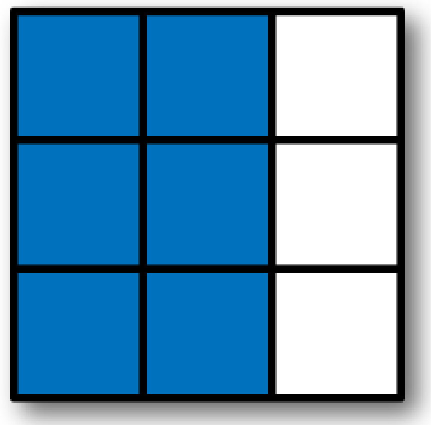
- An interactive, configurable programming environment
- Supports many activities including notebooks
- Runs in your web browser
- Notebooks:
  - Originally designed for Python
  - Supports other languages, too
  - Displays results (even interactive maps) inline
  - You decide how to divide code into executable cells
  - Shift+Enter to execute a cell

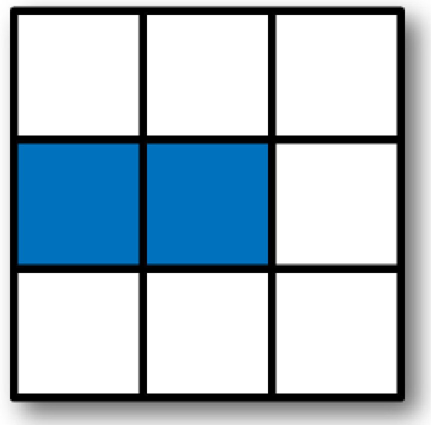
NumPy arrays and slicing



	Expression	Shape
	`arr[:2, 1:]`	(2, 2)

	`arr[2]`
	`arr[2, :]`
	`arr[2:, :]`
		(3,)
		(3,)
		(1, 3)

	`arr[:, :2]`	(3, 2)

	`arr[1, :2]`
	`arr[1:2, :2]`
		(2,)
		(1, 2)

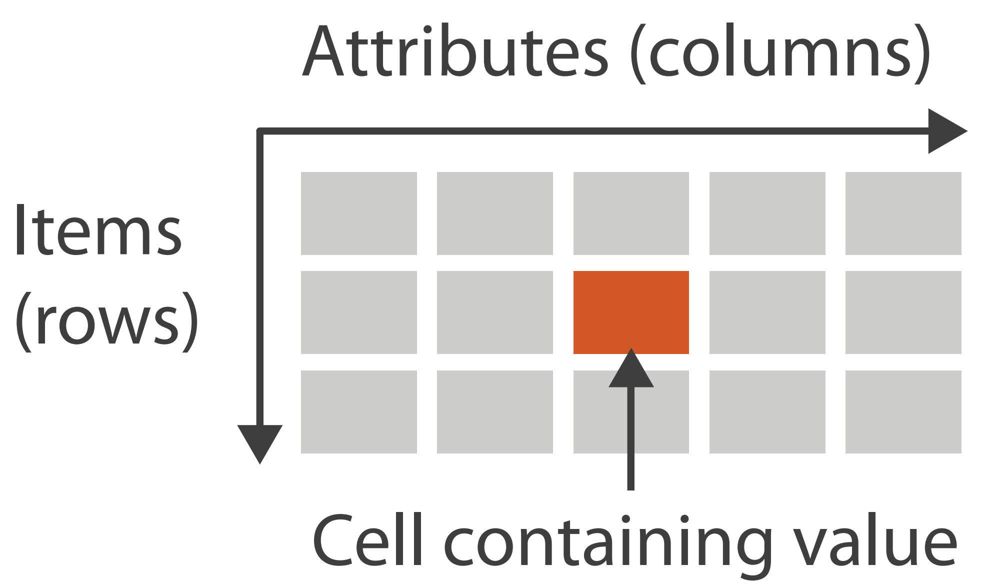
[W. McKinney, Python for Data Analysis]

Boolean Indexing

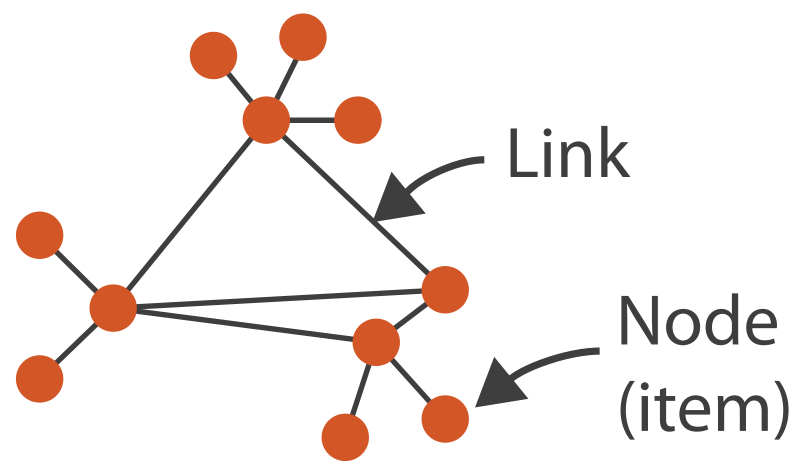
- `names == 'Bob'` gives back booleans that represent the element-wise comparison with the array `names`
- Boolean arrays can be used to index into another array:
  - `data[names == 'Bob']`
- Can even mix and match with integer slicing
- Can do boolean operations (`&`, `|`) between arrays (just like addition, subtraction)
  - `data[(names == 'Bob') | (names == 'Will')]`
- Note: `or` and `and` do not work with arrays
- We can set values too! `data[data < 0] = 0`

What is Data?

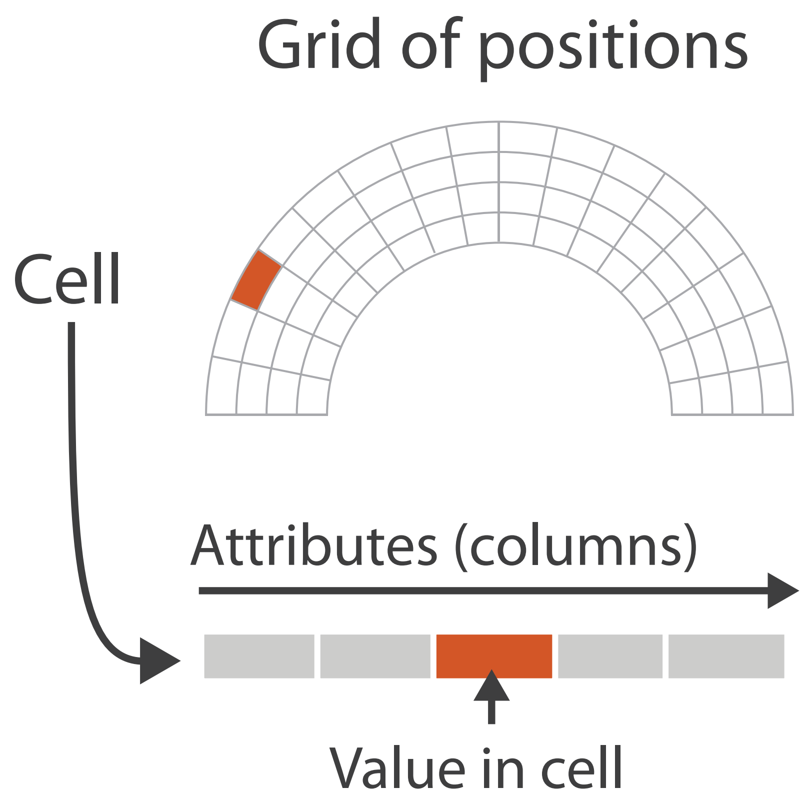
- Tables



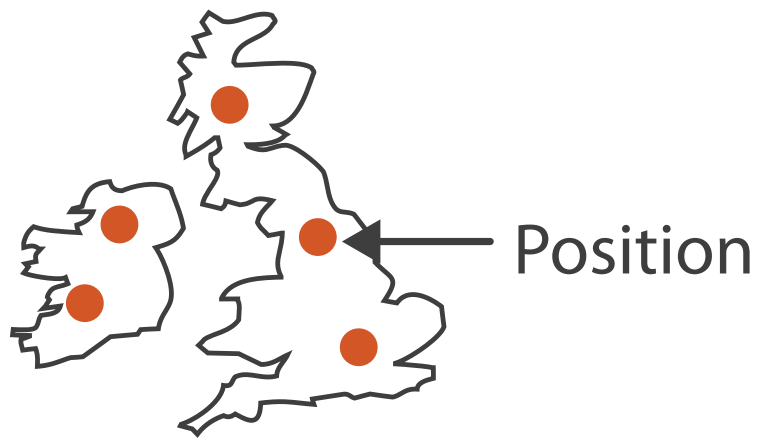
- Networks



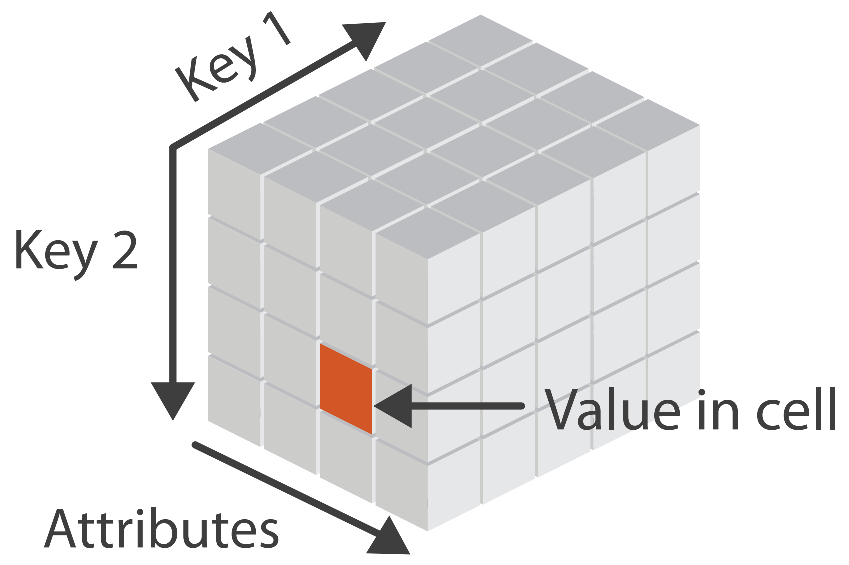
- Fields (Continuous)



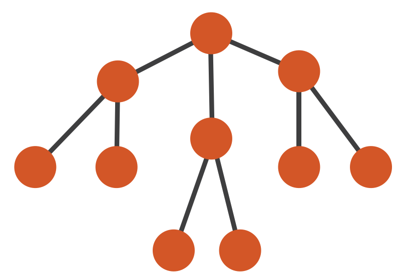
- Geometry (Spatial)



- *Multidimensional Table*

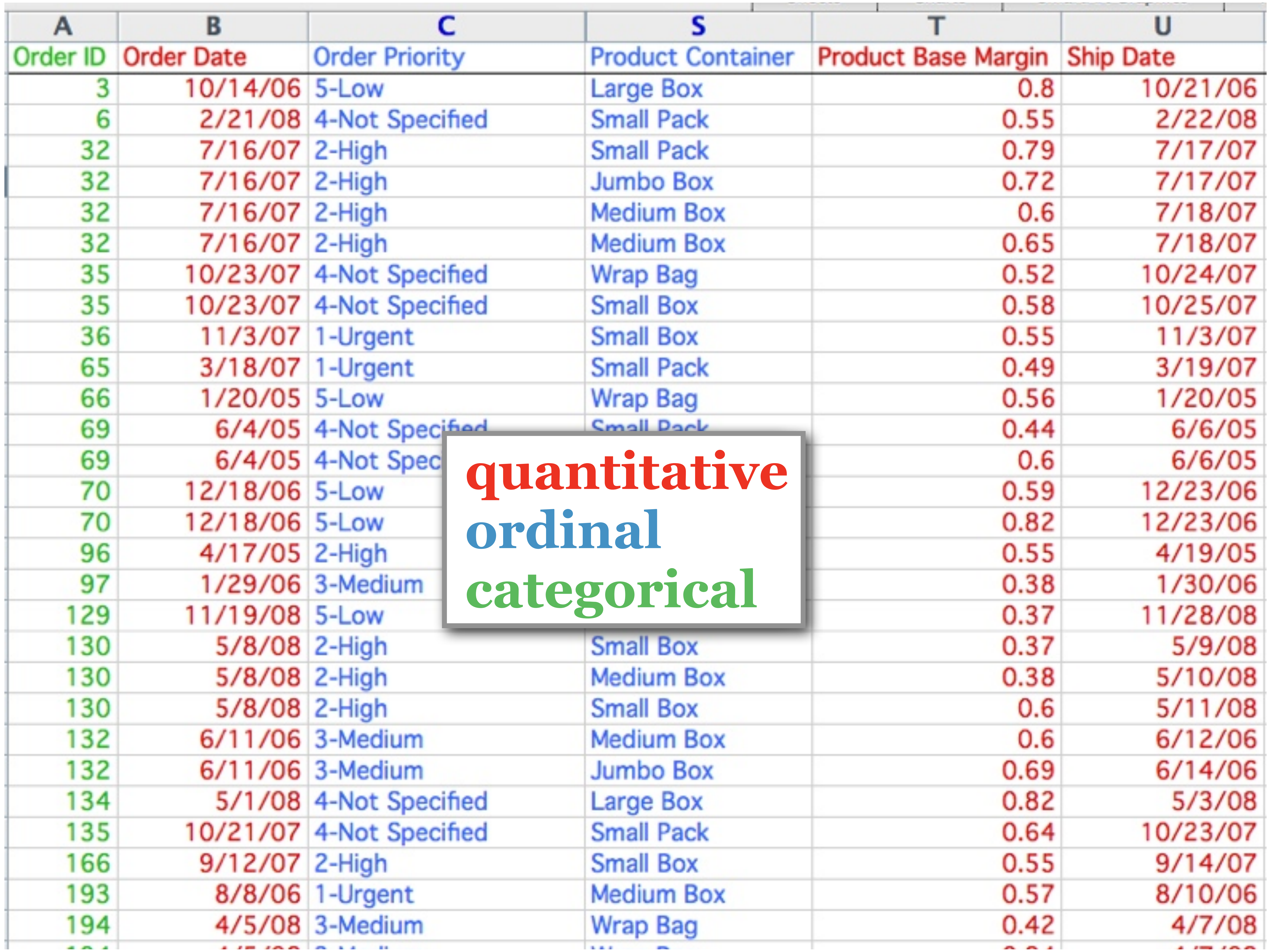


- *Trees*



[Munzner (ill. Maguire), 2014]

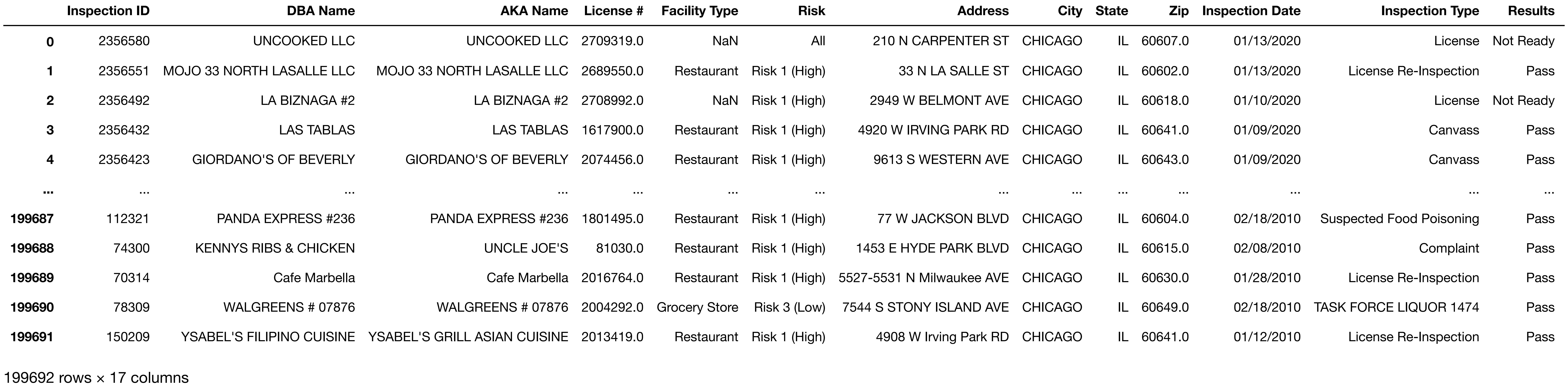
Categorial, Ordinal, and Quantitative



A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box	0.72	7/17/07
32	7/16/07	2-High	Medium Box	0.6	7/18/07
32	7/16/07	2-High	Medium Box	0.65	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66	1/20/05	5-Low	Wrap Bag	0.56	1/20/05
69	6/4/05	4-Not Specified	Small Pack	0.44	6/6/05
69	6/4/05	4-Not Spec		0.6	6/6/05
70	12/18/06	5-Low		0.59	12/23/06
70	12/18/06	5-Low		0.82	12/23/06
96	4/17/05	2-High		0.55	4/19/05
97	1/29/06	3-Medium		0.38	1/30/06
129	11/19/08	5-Low		0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

quantitative
ordinal
categorical

Pandas and Data Frames

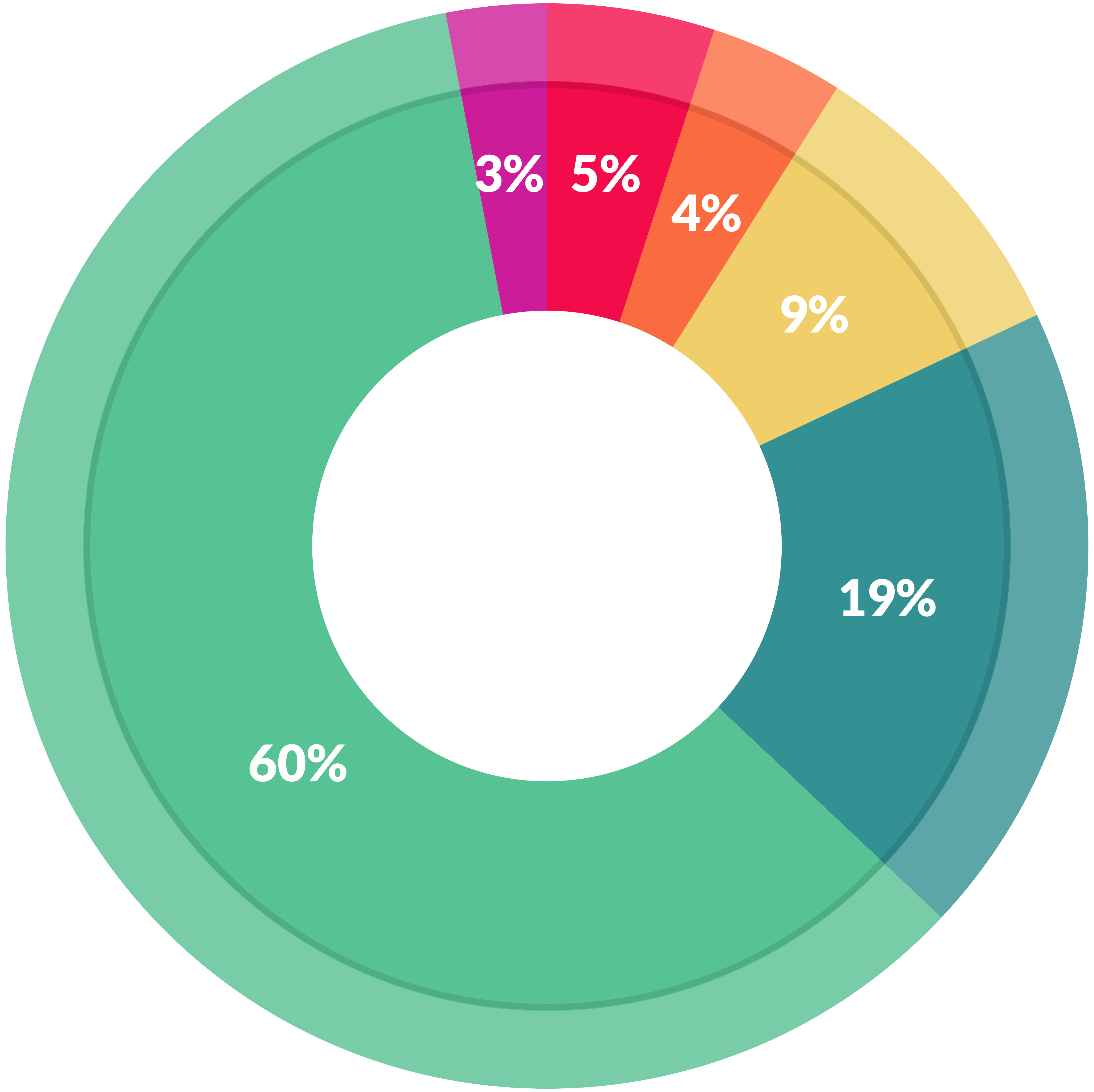


	Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Address	City	State	Zip	Inspection Date	Inspection Type	Results
0	2356580	UNCOOKED LLC	UNCOOKED LLC	2709319.0	NaN	All	210 N CARPENTER ST	CHICAGO	IL	60607.0	01/13/2020	License	Not Ready
1	2356551	MOJO 33 NORTH LASALLE LLC	MOJO 33 NORTH LASALLE LLC	2689550.0	Restaurant	Risk 1 (High)	33 N LA SALLE ST	CHICAGO	IL	60602.0	01/13/2020	License Re-Inspection	Pass
2	2356492	LA BIZNAGA #2	LA BIZNAGA #2	2708992.0	NaN	Risk 1 (High)	2949 W BELMONT AVE	CHICAGO	IL	60618.0	01/10/2020	License	Not Ready
3	2356432	LAS TABLAS	LAS TABLAS	1617900.0	Restaurant	Risk 1 (High)	4920 W IRVING PARK RD	CHICAGO	IL	60641.0	01/09/2020	Canvass	Pass
4	2356423	GIORDANO'S OF BEVERLY	GIORDANO'S OF BEVERLY	2074456.0	Restaurant	Risk 1 (High)	9613 S WESTERN AVE	CHICAGO	IL	60643.0	01/09/2020	Canvass	Pass
...	...	...	...	...	...	...	...	...	...	...	...	...	...
199687	112321	PANDA EXPRESS #236	PANDA EXPRESS #236	1801495.0	Restaurant	Risk 1 (High)	77 W JACKSON BLVD	CHICAGO	IL	60604.0	02/18/2010	Suspected Food Poisoning	Pass
199688	74300	KENNYS RIBS & CHICKEN	UNCLE JOE'S	81030.0	Restaurant	Risk 1 (High)	1453 E HYDE PARK BLVD	CHICAGO	IL	60615.0	02/08/2010	Complaint	Pass
199689	70314	Cafe Marbella	Cafe Marbella	2016764.0	Restaurant	Risk 1 (High)	5527-5531 N Milwaukee AVE	CHICAGO	IL	60630.0	01/28/2010	License Re-Inspection	Pass
199690	78309	WALGREENS # 07876	WALGREENS # 07876	2004292.0	Grocery Store	Risk 3 (Low)	7544 S STONY ISLAND AVE	CHICAGO	IL	60649.0	02/18/2010	TASK FORCE LIQUOR 1474	Pass
199691	150209	YSABEL'S FILIPINO CUISINE	YSABEL'S GRILL ASIAN CUISINE	2013419.0	Restaurant	Risk 1 (High)	4908 W Irving Park RD	CHICAGO	IL	60641.0	01/12/2010	License Re-Inspection	Pass

199692 rows × 17 columns

- Data Frames are tables with many database-like operations
- Index shared across all columns
- Can select, project, merge (join), and more
- Read and write many file formats

How do data scientists spend their time?

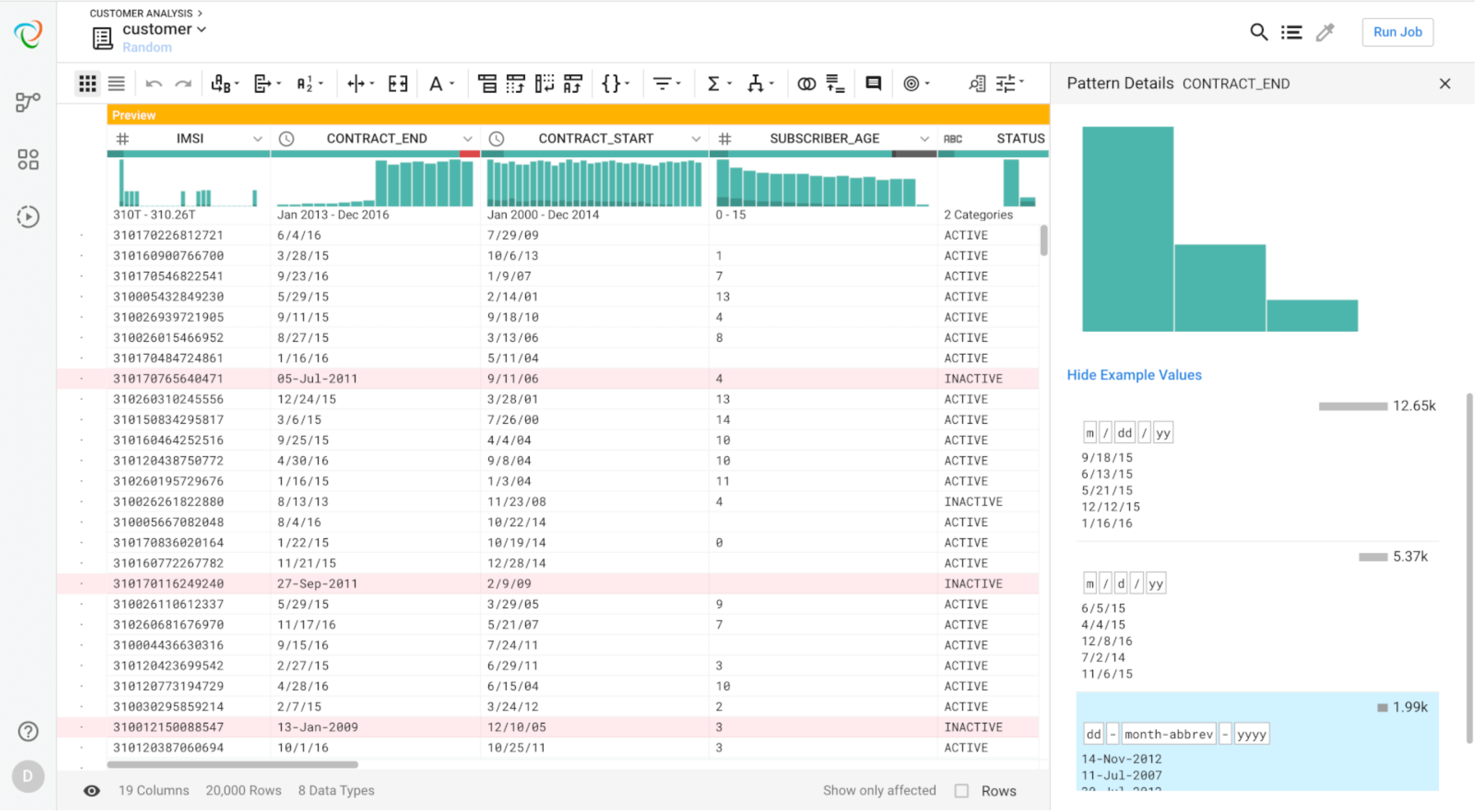


What data scientists spend the most time doing

- ● *Building training sets: 3%*
- ● *Cleaning and organizing data: 60%*
- ● *Collecting data sets; 19%*
- ● *Mining data for patterns: 9%*
- ● *Refining algorithms: 4%*
- ● *Other: 5%*

[CrowdFlower Data Science Report, 2016]

Data Wrangling



IMSI	CONTRACT_END	CONTRACT_START	SUBSCRIBER_AGE	STATUS
310T - 310.26T	Jan 2013 - Dec 2016	Jan 2000 - Dec 2014	0 - 15	2 Categories
310170226812721	6/4/16	7/29/09		ACTIVE
310160900766700	3/28/15	10/6/13	1	ACTIVE
310170546822541	9/23/16	1/9/07	7	ACTIVE
310005432849230	5/29/15	2/14/01	13	ACTIVE
310026939721905	9/11/15	9/18/10	4	ACTIVE
310026015466952	8/27/15	3/13/06	8	ACTIVE
310170484724861	1/16/16	5/11/04		ACTIVE
310170765640471	05-Jul-2011	9/11/06	4	INACTIVE
310260310245556	12/24/15	3/28/01	13	ACTIVE
310150834295817	3/6/15	7/26/00	14	ACTIVE
310160464252516	9/25/15	4/4/04	10	ACTIVE
310120438750772	4/30/16	9/8/04	10	ACTIVE
310260195729676	1/16/15	1/3/04	11	ACTIVE
310026261822880	8/13/13	11/23/08	4	INACTIVE
310005667082048	8/4/16	10/22/14		ACTIVE
310170836020164	1/22/15	10/19/14	0	ACTIVE
310160772267782	11/21/15	12/28/14		ACTIVE
310170116249240	27-Sep-2011	2/9/09		INACTIVE
310026110612337	5/29/15	3/29/05	9	ACTIVE
310260681676970	11/17/16	5/21/07	7	ACTIVE
310004436630316	9/15/16	7/24/11		ACTIVE
310120423699542	2/27/15	6/29/11	3	ACTIVE
310120773194729	4/28/16	6/15/04	10	ACTIVE
310030295859214	2/7/15	3/24/12	2	ACTIVE
310012150088547	13-Jan-2009	12/10/05	3	INACTIVE
310120387060694	10/1/16	10/25/11	3	ACTIVE

19 Columns 20,000 Rows 8 Data Types

Show only affected □ Rows

Pattern Details CONTRACT_END

Hide Example Values

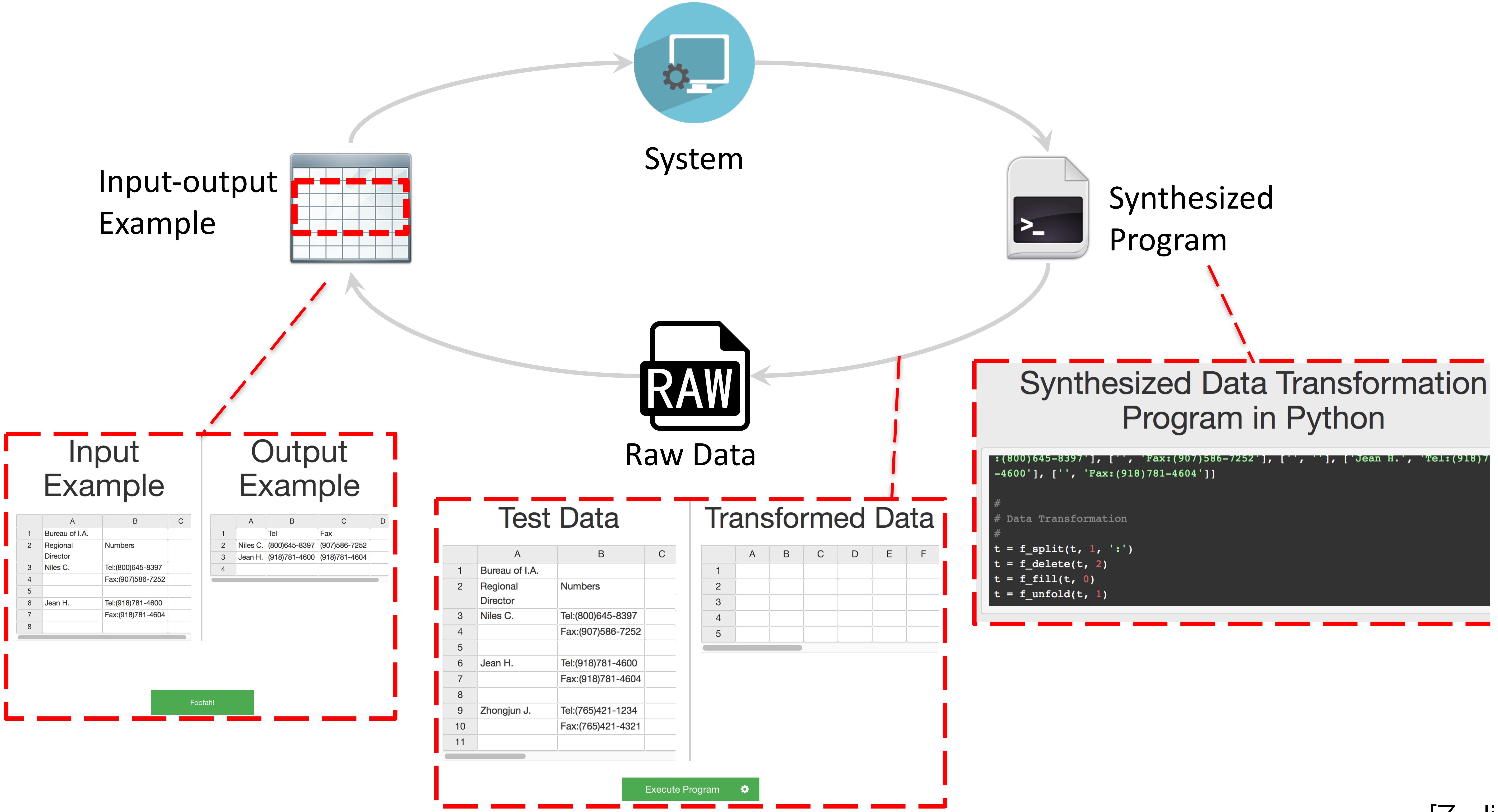
12.65k
m / dd / yy
9/18/15
6/13/15
5/21/15
12/12/15
1/16/16

5.37k
m / d / yy
6/5/15
4/4/15
12/8/16
7/2/14
11/6/15

1.99k
dd - month-abbrev - yyyy
14-Nov-2012
11-Jul-2007

[Trifacta]

Foofah: Programming by Example



Input Example

	A	B	C
1	Bureau of I.A.		
2	Regional Director	Numbers	
3	Niles C.	Tel:(800)645-8397	
4		Fax:(907)586-7252	
5			
6	Jean H.	Tel:(918)781-4600	
7		Fax:(918)781-4604	
8			

Output Example

	A	B	C	D
1		Tel	Fax	
2	Niles C.	(800)645-8397	(907)586-7252	
3	Jean H.	(918)781-4600	(918)781-4604	
4				

Test Data

	A	B	C
1	Bureau of I.A.		
2	Regional Director	Numbers	
3	Niles C.	Tel:(800)645-8397	
4		Fax:(907)586-7252	
5			
6	Jean H.	Tel:(918)781-4600	
7		Fax:(918)781-4604	
8			
9	Zhongjun J.	Tel:(765)421-1234	
10		Fax:(765)421-4321	
11			

Transformed Data

	A	B	C	D	E	F
1						
2						
3						
4						
5						

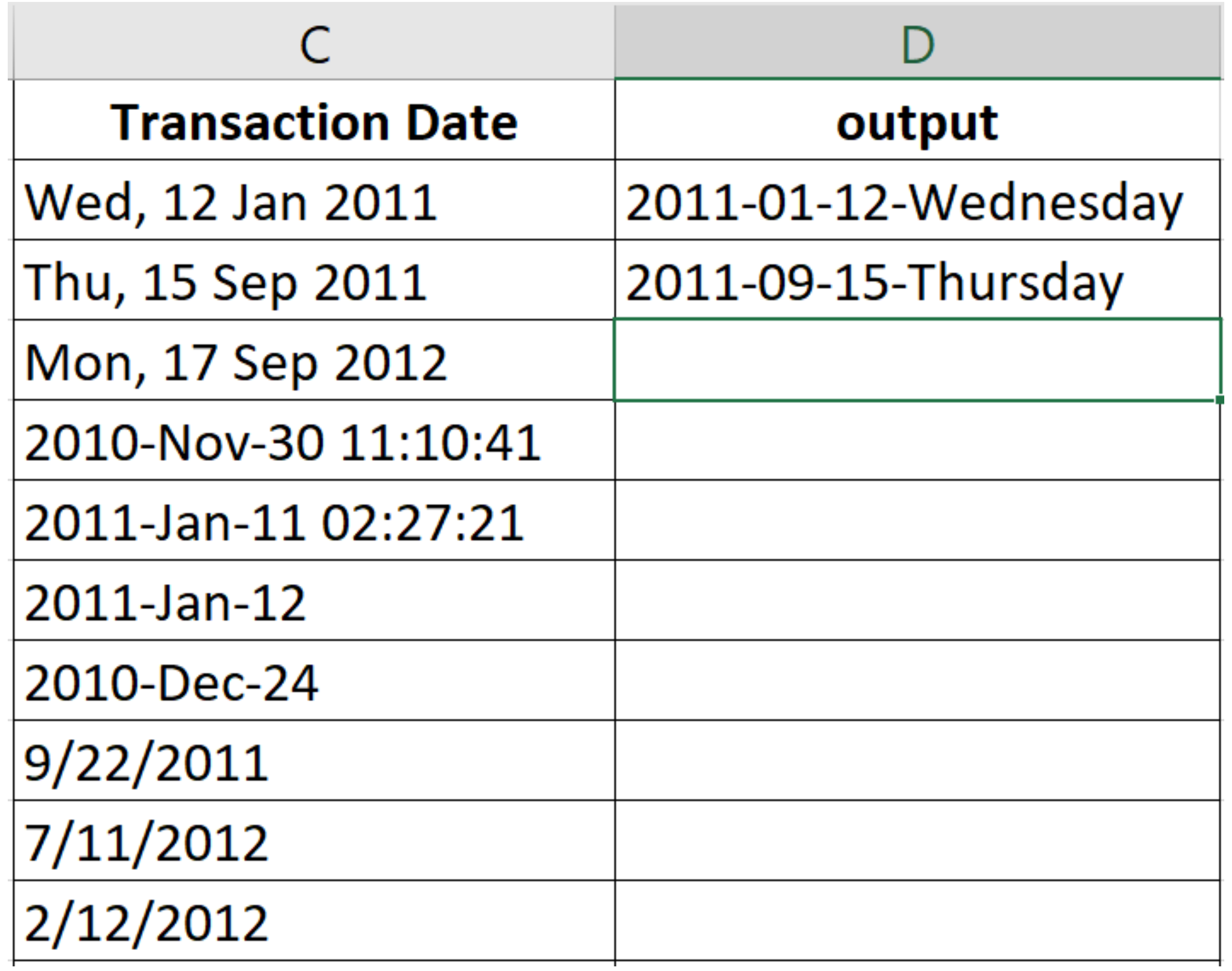
Synthesized Data Transformation Program in Python

```
:(800)645-8397'], ['', 'Fax:(907)586-7252'], ['', ''], ['Jean H.', 'Tel:(918)7
-4600'], ['', 'Fax:(918)781-4604']]

#
# Data Transformation
#
t = f_split(t, 1, ':')
t = f_delete(t, 2)
t = f_fill(t, 0)
t = f_unfold(t, 1)
```

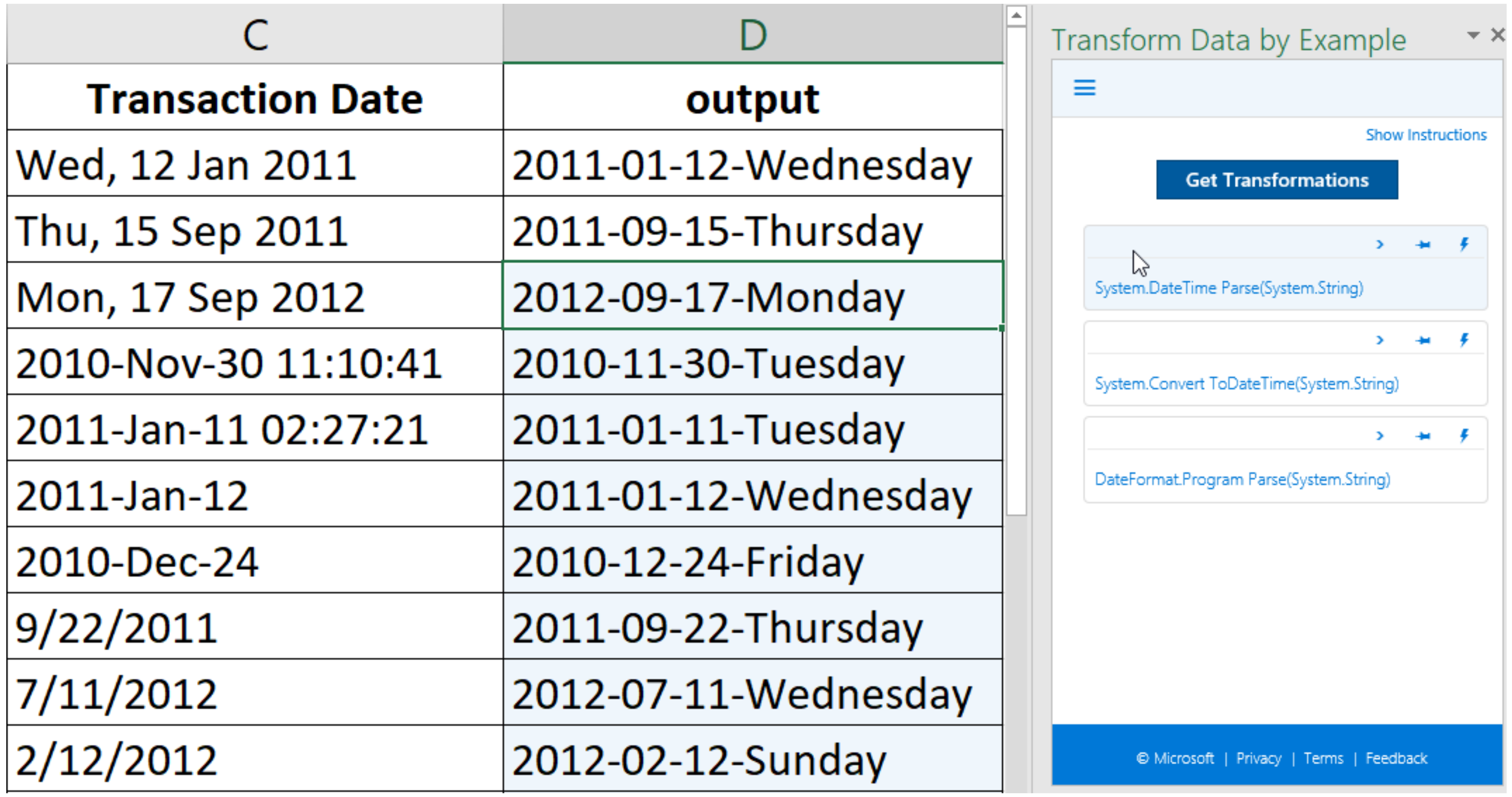
[Z. Jin et al., 2017]

TDE: Transform Data by Example



C	D
Transaction Date	**output**
Wed, 12 Jan 2011	2011-01-12-Wednesday
Thu, 15 Sep 2011	2011-09-15-Thursday
Mon, 17 Sep 2012	
2010-Nov-30 11:10:41	
2011-Jan-11 02:27:21	
2011-Jan-12	
2010-Dec-24	
9/22/2011	
7/11/2012	
2/12/2012	

→



C	D
Transaction Date	**output**
Wed, 12 Jan 2011	2011-01-12-Wednesday
Thu, 15 Sep 2011	2011-09-15-Thursday
Mon, 17 Sep 2012	2012-09-17-Monday
2010-Nov-30 11:10:41	2010-11-30-Tuesday
2011-Jan-11 02:27:21	2011-01-11-Tuesday
2011-Jan-12	2011-01-12-Wednesday
2010-Dec-24	2010-12-24-Friday
9/22/2011	2011-09-22-Thursday
7/11/2012	2012-07-11-Wednesday
2/12/2012	2012-02-12-Sunday

Transform Data by Example
Show Instructions
Get Transformations
System.DateTime Parse(System.String)
System.Convert ToDateTime(System.String)
DateFormat.Program Parse(System.String)
© Microsoft | Privacy | Terms | Feedback

[Y. He et al., 2018]

Tidy Data

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

Initial Data

	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

Transpose

name	trt	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Tidy Data

[H. Wickham, 2014]

MultiIndex Row Access and Slicing

- `df.loc[("Boston", 2007)]` or sometimes `df.loc["Boston", 2007]`
- Remember that `loc` uses the index values, `iloc` uses integers
- **Note**: `df.iloc[0]` gets the first row, **not** `df.iloc[0,0]`
- Can get a subset of the data using partial indices
  - `df.loc["Boston"]` returns both 2007 and 2008 data
- What about slicing?
  - `df.loc["Boston":"Cleveland"]` → ERROR! (Need sorted data)
  - `df = df.sort_index()`
  - `df.loc["Boston":"Cleveland"]` → inclusive!
  - `df.loc[(slice("Boston","Cleveland"),2007),:]`

Merges (aka Joins)

- Need to merge data from one DataFrame with data from another DataFrame
- Example: Football game data merged with temperature data

Game

Id	Location	Date	Home	Away
0	Boston	9/2	1	15
1	Boston	9/9	1	7
2	Cleveland	9/16	12	1
3	San Diego	9/23	21	1

Weather

wId	City	Date	Temp
0	Boston	9/2	72
1	Boston	9/3	68
…	…	…	…
7	Boston	9/9	75
…	…	…	…
21	Boston	9/23	54
…	…	…	…
36	Cleveland	9/16	81

No data for San Diego

Inner Strategy

Merged

Id	Location	Date	Home	Away	Temp	wId
0	Boston	9/2	1	15	72	0
1	Boston	9/9	1	7	75	7
2	Cleveland	9/16	12	1	81	36

No San Diego entry

Outer Strategy

Merged

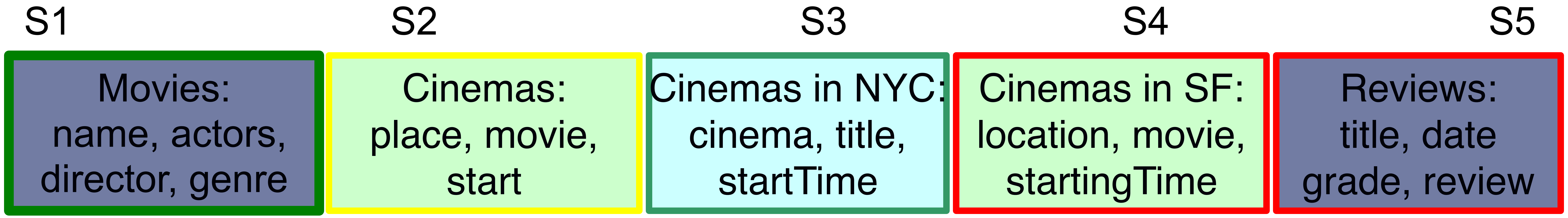
Id	Location	Date	Home	Away	Temp	wId
0	Boston	9/2	1	15	72	0
NaN	Boston	9/3	NaN	NaN	68	1
…	…	…	…	…	…	…
1	Boston	9/9	1	7	75	7
NaN	Boston	9/10	NaN	NaN	76	8
…	…	…	…	…	…	…
NaN	Cleveland	9/2	NaN	NaN	61	22
…	…	…	…	…	…	…
2	Cleveland	9/16	12	1	81	36
…	…	…	…	…	…	…
3	San Diego	9/23	21	1	NaN	NaN

Data Integration

```
select title, startTime
from Movie, Plays
where Movie.title=Plays.movie AND
          location="New York"  AND
          director="Woody Allen"
```

Movie: Title, director, year, genre
Actors: title, actor
Plays: movie, location, startTime
Reviews: title, rating, description

Sources S1 and S3 are relevant, sources S4 and S5 are irrelevant, and source S2 is relevant but possibly redundant.



S1
Movies:
name, actors,
director, genre

S2
Cinemas:
place, movie,
start

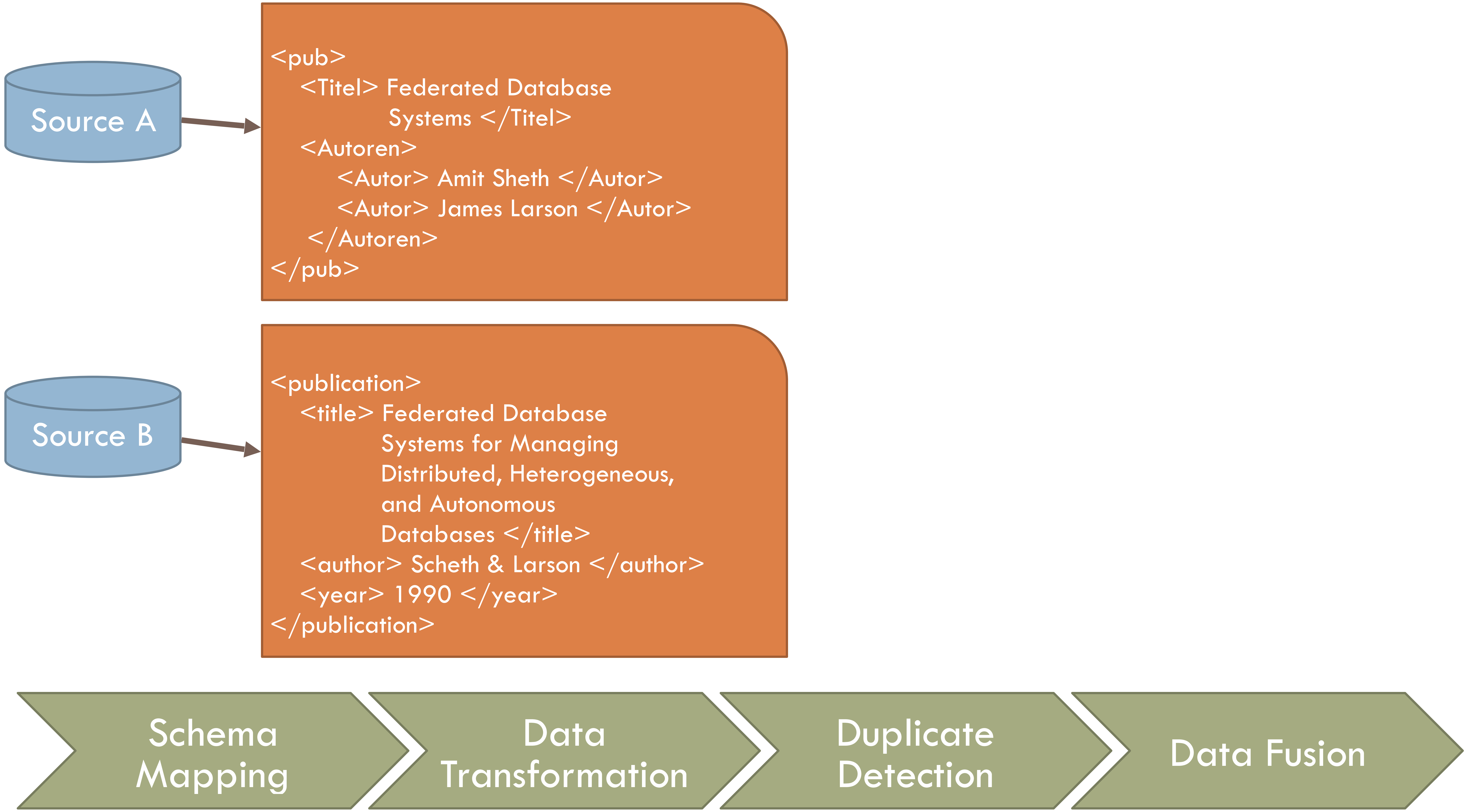
S3
Cinemas in NYC:
cinema, title,
startTime

S4
Cinemas in SF:
location, movie,
startingTime

S5
Reviews:
title, date
grade, review

[AH Doan et al., 2012]

Information Integration

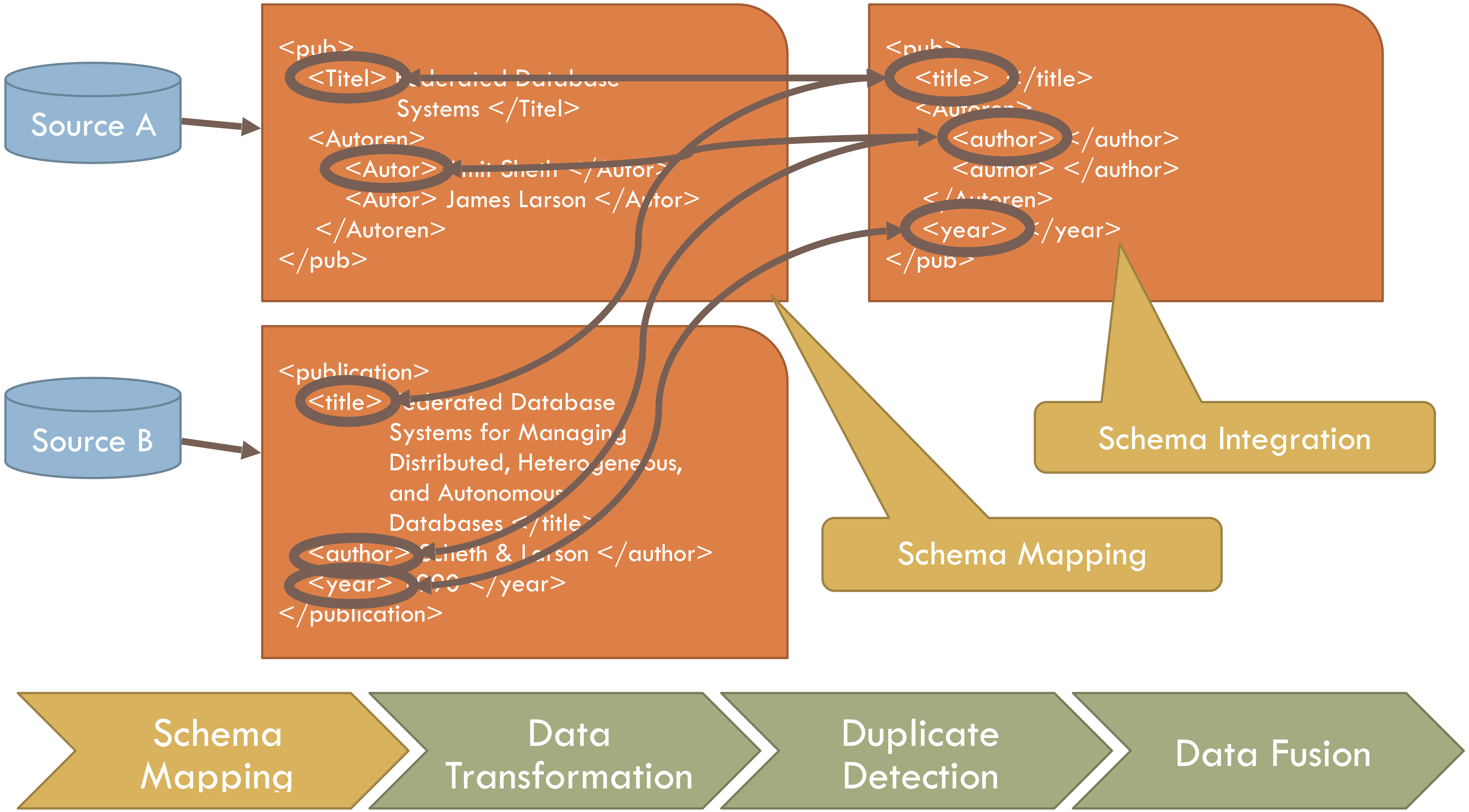


```
<pub>
   <Titel> Federated Database
            Systems </Titel>
   <Autoren>
       <Autor> Amit Sheth </Autor>
       <Autor> James Larson </Autor>
    </Autoren>
</pub>
```

```
<publication>
   <title> Federated Database
           Systems for Managing
           Distributed, Heterogeneous,
           and Autonomous
           Databases </title>
   <author> Scheth & Larson </author>
   <year> 1990 </year>
</publication>
```

[L. Dong and F. Naumann, 2009]

Information Integration



Source A

```
<pub>
  <Titel> Federated Database
          Systems </Titel>
  <Autoren>
    <Autor> Amit Sheth </Autor>
    <Autor> James Larson </Autor>
  </Autoren>
</pub>
```

Source B

```
<publication>
  <title> Federated Database
         Systems for Managing
         Distributed, Heterogeneous,
         and Autonomous
         Databases </title>
  <author> Sheth & Larson </author>
  <year> 1990 </year>
</publication>
```

```
<pub>
  <title> </title>
  <Autoren>
    <author> </author>
    <author> </author>
  </Autoren>
  <year> </year>
</pub>
```

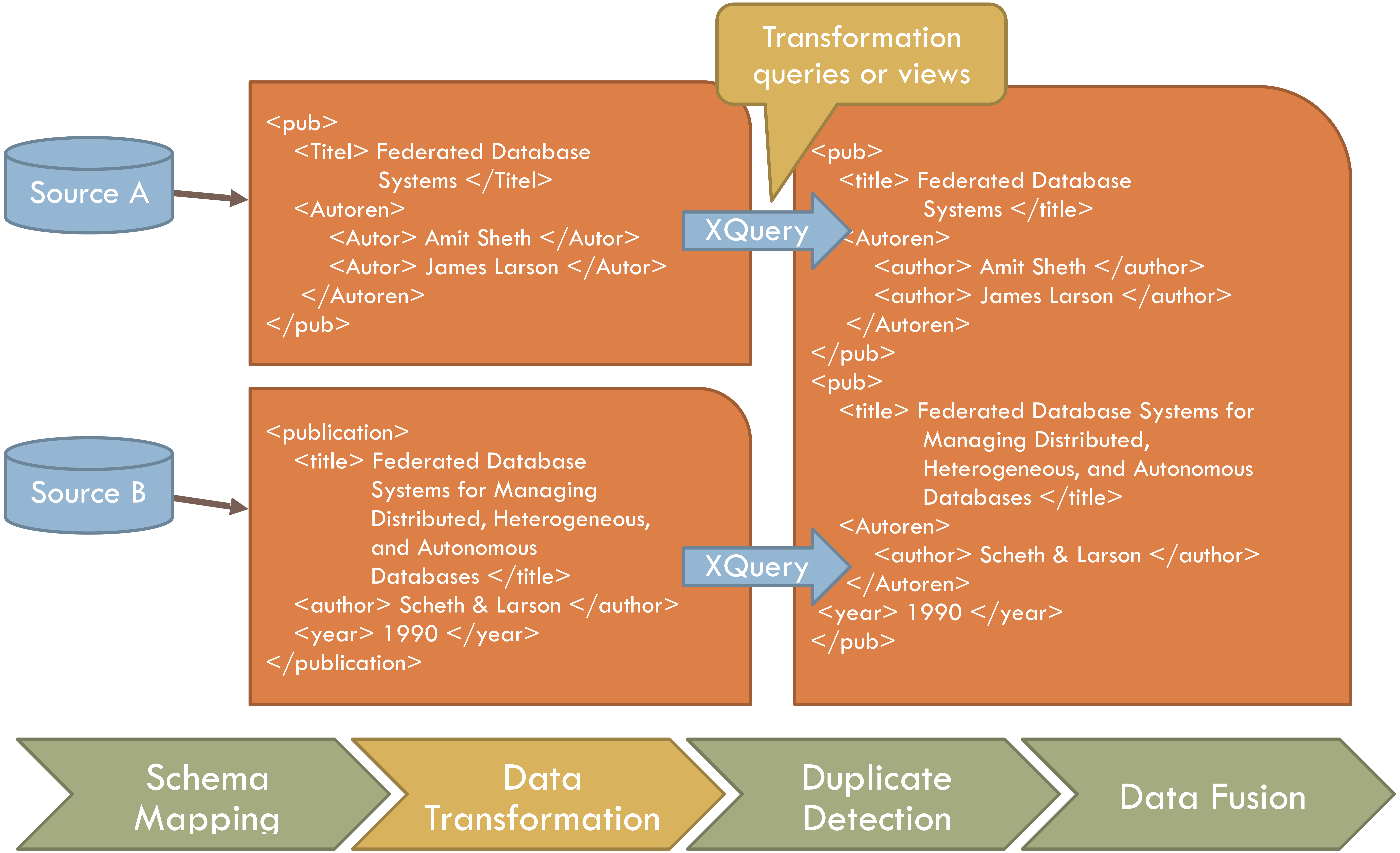
Schema Integration

Schema Mapping

Schema Mapping → Data Transformation → Duplicate Detection → Data Fusion

[L. Dong and F. Naumann, 2009]

Information Integration



Transformation queries or views

Source A

```
<pub>
   <Titel> Federated Database
            Systems </Titel>
   <Autoren>
       <Autor> Amit Sheth </Autor>
       <Autor> James Larson </Autor>
    </Autoren>
</pub>
```

Source B

```
<publication>
   <title> Federated Database
            Systems for Managing
            Distributed, Heterogeneous,
            and Autonomous
            Databases </title>
   <author> Scheth & Larson </author>
   <year> 1990 </year>
</publication>
```

XQuery

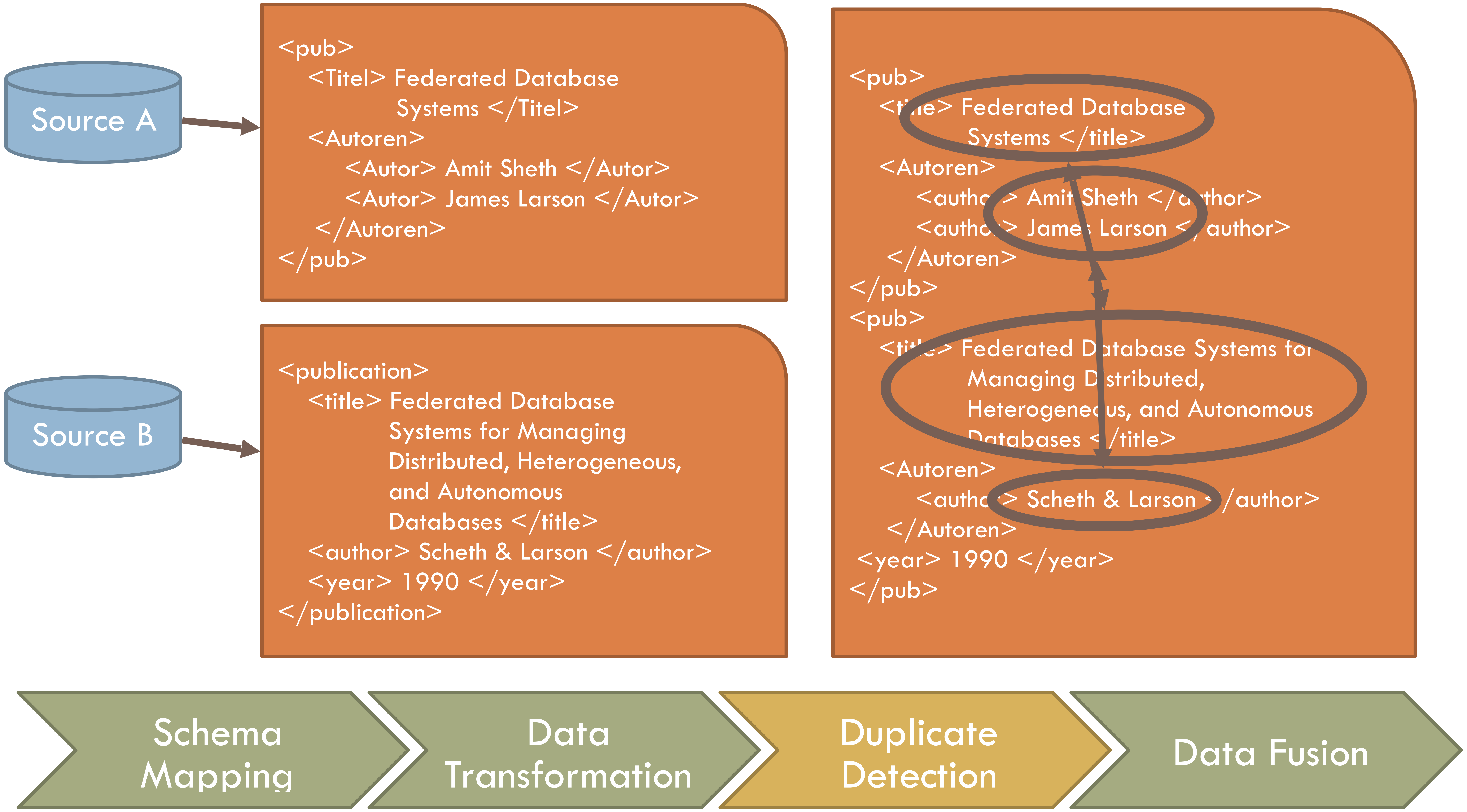
XQuery

```
<pub>
   <title> Federated Database
            Systems </title>
   <Autoren>
       <author> Amit Sheth </author>
       <author> James Larson </author>
    </Autoren>
</pub>
<pub>
   <title> Federated Database Systems for
            Managing Distributed,
            Heterogeneous, and Autonomous
            Databases </title>
   <Autoren>
       <author> Scheth & Larson </author>
    </Autoren>
  <year> 1990 </year>
</pub>
```

Schema Mapping → Data Transformation → Duplicate Detection → Data Fusion

[L. Dong and F. Naumann, 2009]

Information Integration



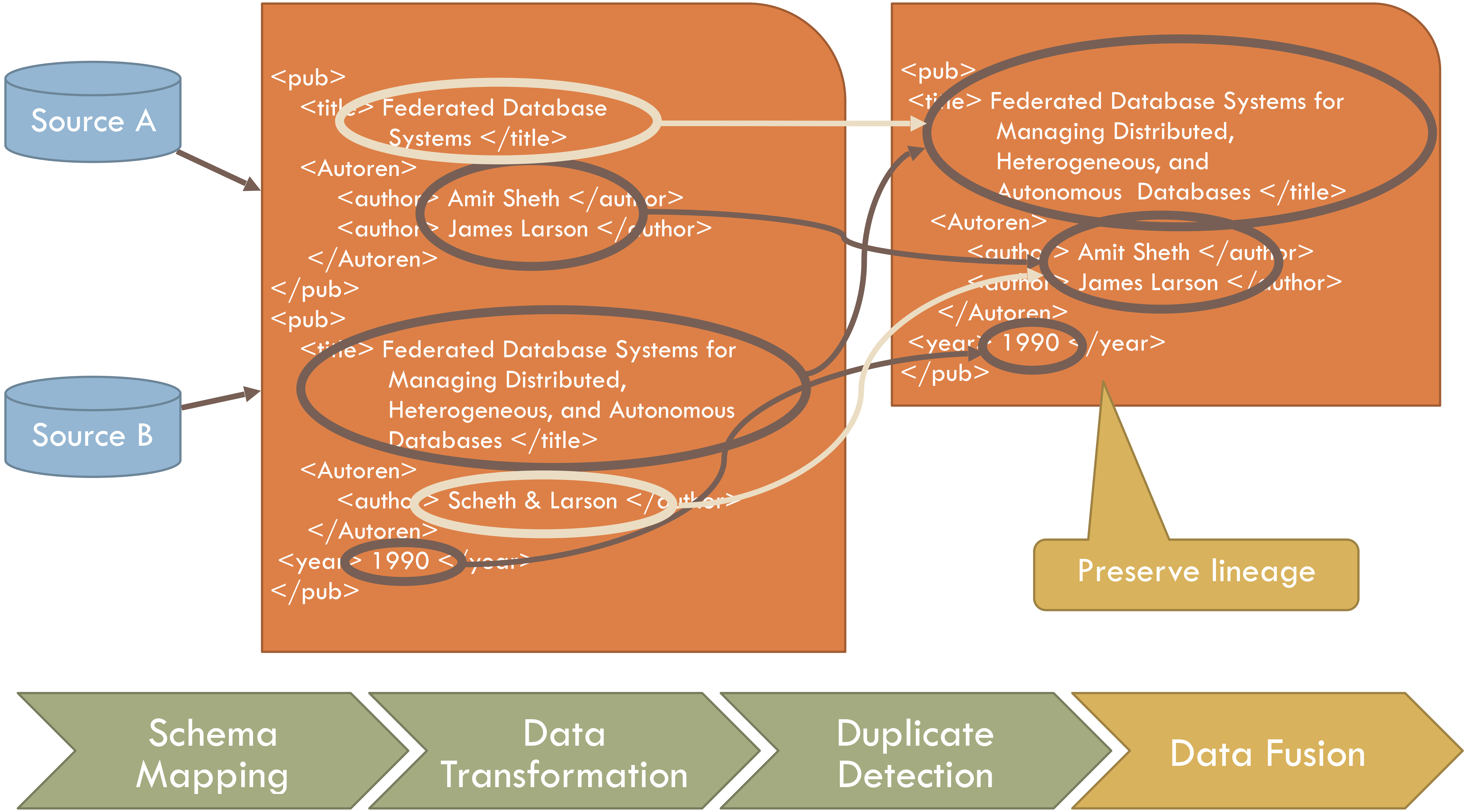
```
<pub>
   <Titel> Federated Database
             Systems </Titel>
   <Autoren>
       <Autor> Amit Sheth </Autor>
       <Autor> James Larson </Autor>
    </Autoren>
</pub>
```

```
<publication>
   <title> Federated Database
            Systems for Managing
            Distributed, Heterogeneous,
            and Autonomous
            Databases </title>
   <author> Scheth & Larson </author>
   <year> 1990 </year>
</publication>
```

```
<pub>
   <title> Federated Database
            Systems </title>
   <Autoren>
       <author> Amit Sheth </author>
       <author> James Larson </author>
    </Autoren>
</pub>
<pub>
   <title> Federated Database Systems for
            Managing Distributed,
            Heterogeneous, and Autonomous
            Databases </title>
   <Autoren>
       <author> Scheth & Larson </author>
    </Autoren>
  <year> 1990 </year>
</pub>
```

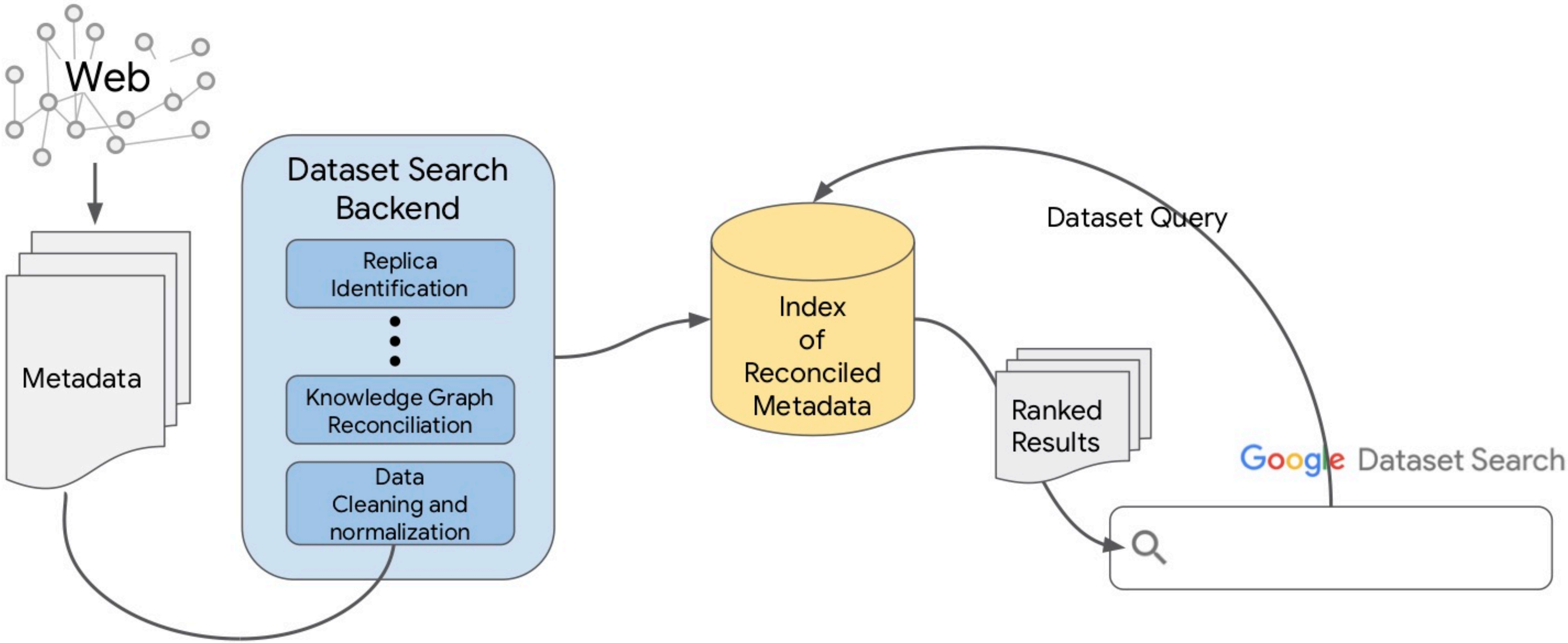
[L. Dong and F. Naumann, 2009]

Information Integration


Source A
Source B
<pub>
<title> Federated Database Systems </title>
<Autoren>
<author> Amit Sheth </author>
<author> James Larson </author>
</Autoren>
</pub>
<pub>
<title> Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases </title>
<Autoren>
<author> Scheth & Larson </author>
</Autoren>
<year> 1990 </year>
</pub>
<pub>
<title> Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases </title>
<Autoren>
<author> Amit Sheth </author>
<author> James Larson </author>
</Autoren>
<year> 1990 </year>
</pub>
Preserve lineage
Schema Mapping
Data Transformation
Duplicate Detection
Data Fusion

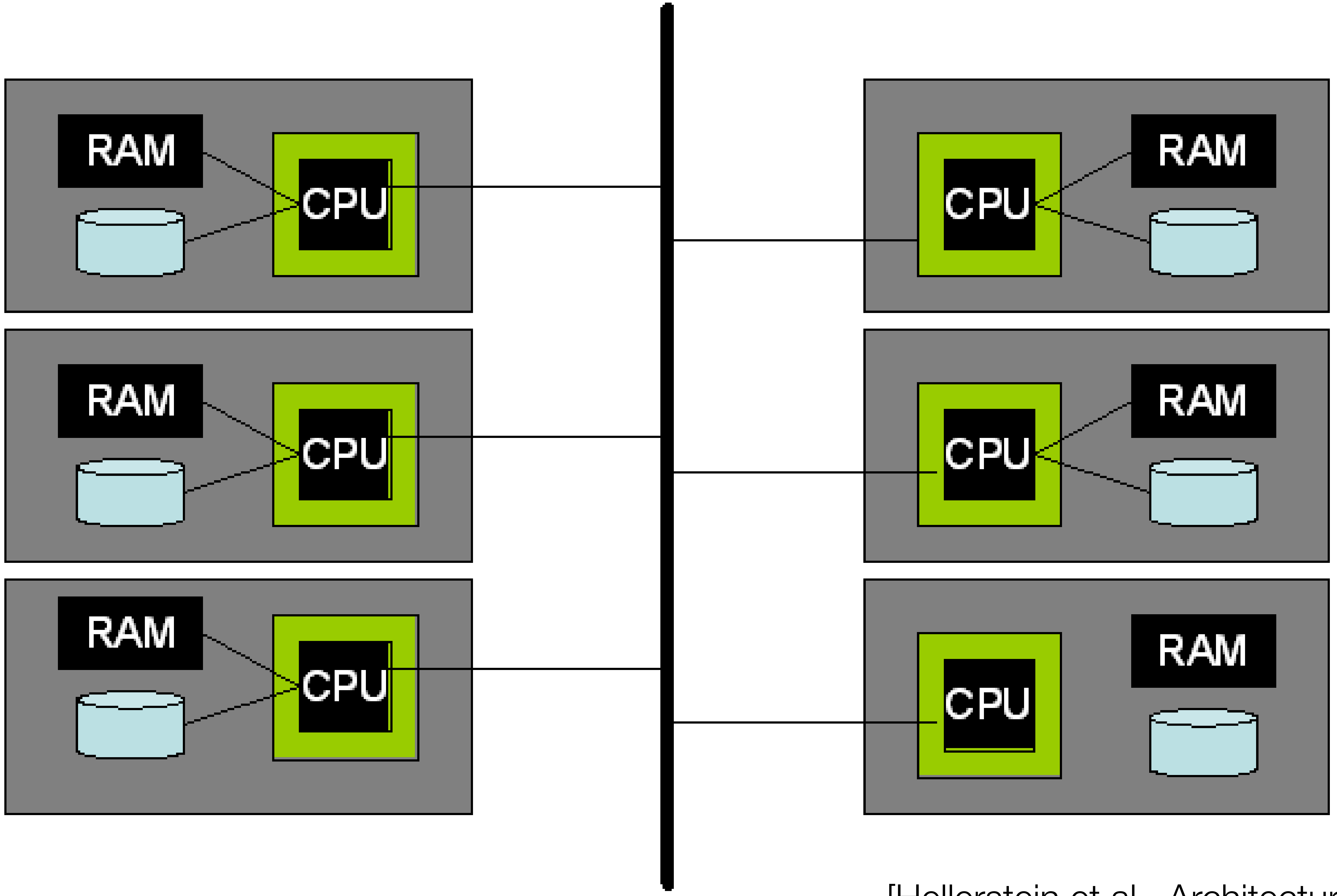
[L. Dong and F. Naumann, 2009]

Google Dataset Search Overview



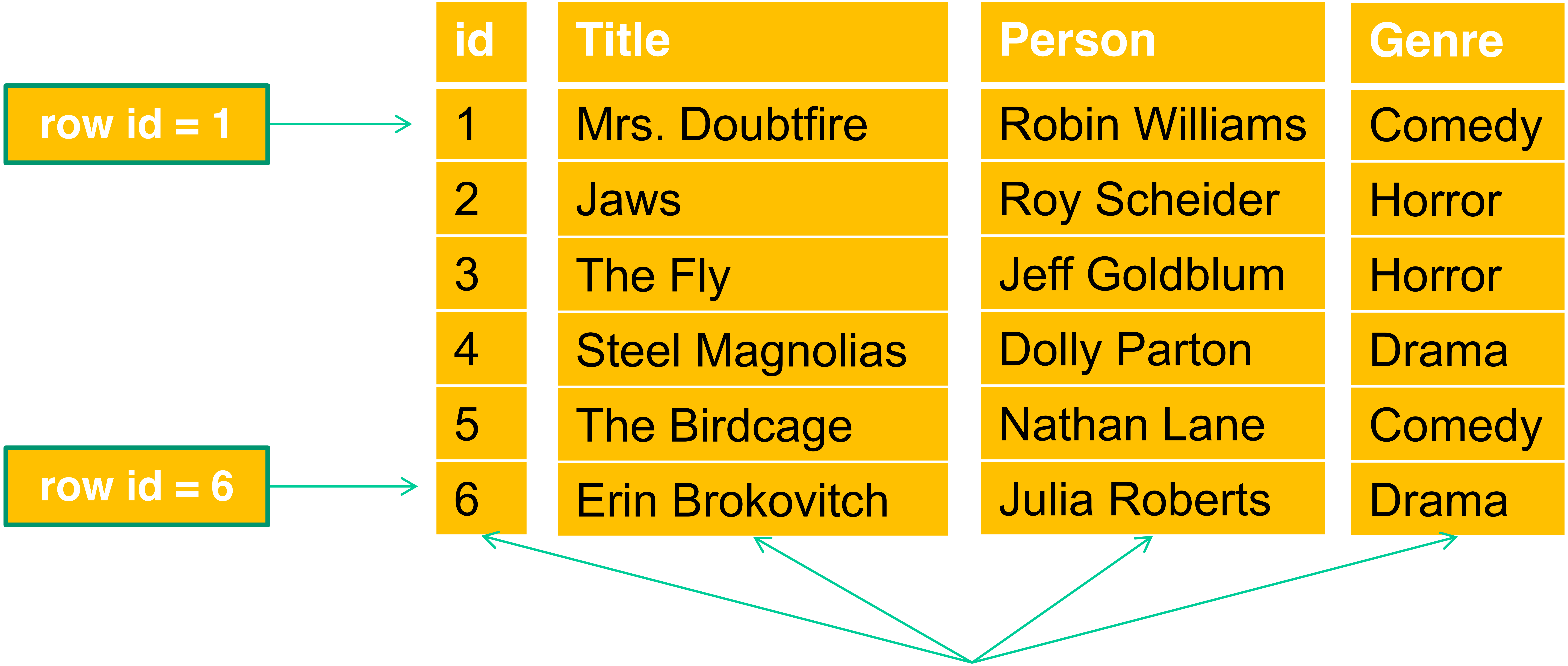
[N. Noy et al., 2019]

Parallel DB Architecture: Shared Nothing



[Hellerstein et al., Architecture of a Database System]

Column Stores



row id = 1

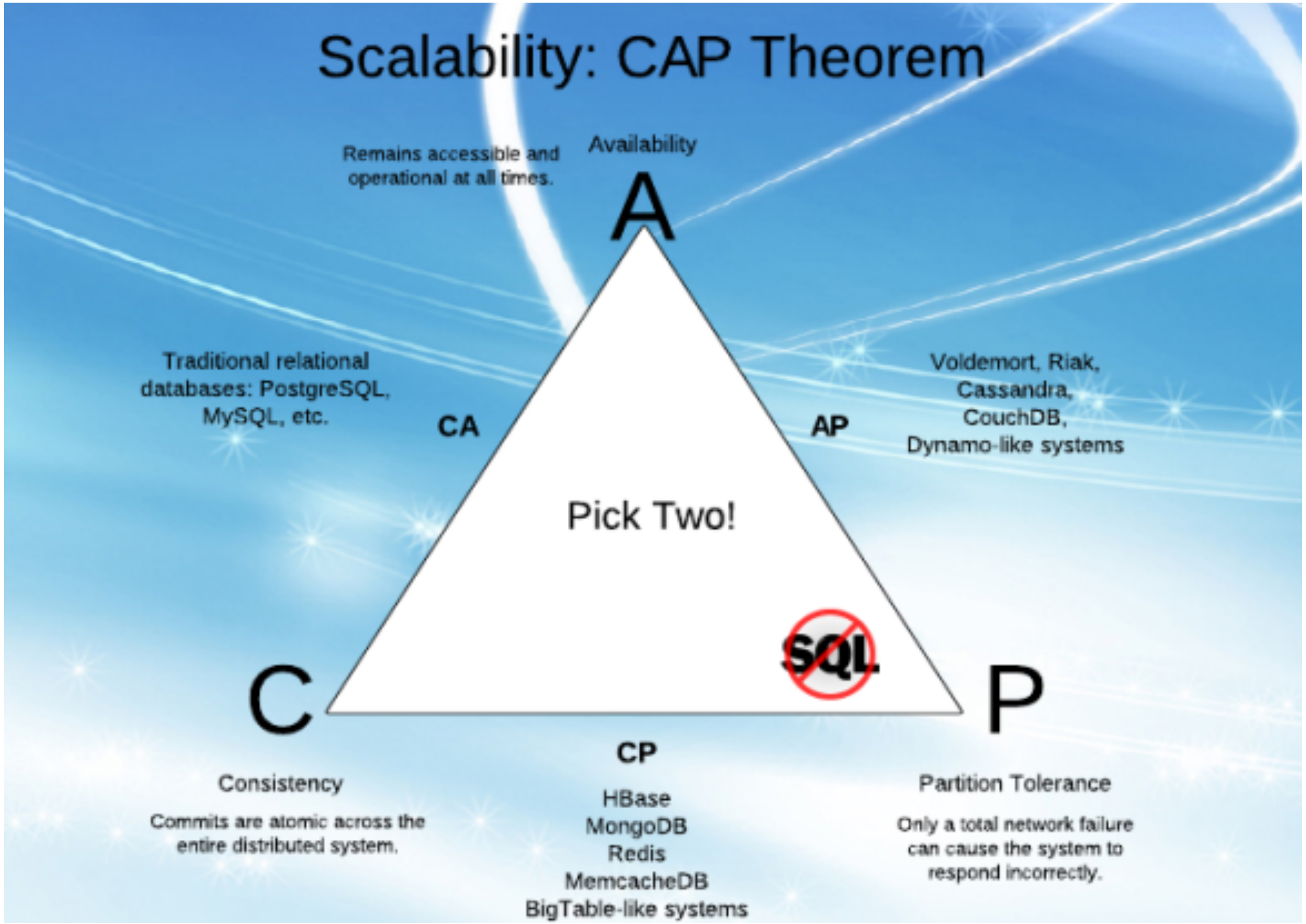
row id = 6

id	Title	Person	Genre
1	Mrs. Doubtfire	Robin Williams	Comedy
2	Jaws	Roy Scheider	Horror
3	The Fly	Jeff Goldblum	Horror
4	Steel Magnolias	Dolly Parton	Drama
5	The Birdcage	Nathan Lane	Comedy
6	Erin Brokovitch	Julia Roberts	Drama

Each column has a file or segment on disk

[J. Swanhart, Introduction to Column Stores]

CAP Theorem



Scalability: CAP Theorem

Remains accessible and operational at all times.

Availability

A

Traditional relational databases: PostgreSQL, MySQL, etc.

CA

AP

Voldemort, Riak, Cassandra, CouchDB, Dynamo-like systems

Pick Two!

SQL

C

P

CP

Consistency

Commits are atomic across the entire distributed system.

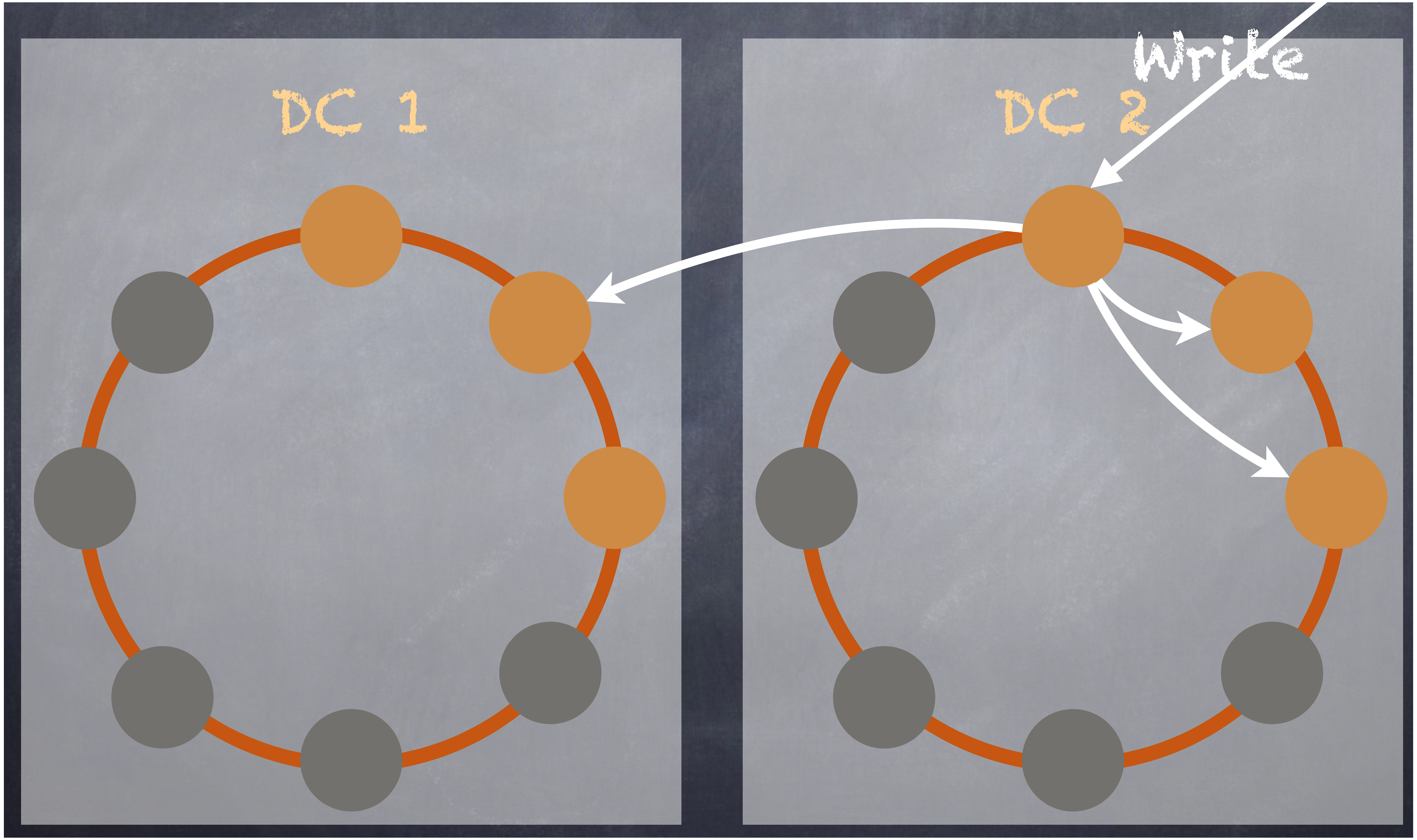
HBase
MongoDB
Redis
MemcacheDB
BigTable-like systems

Partition Tolerance

Only a total network failure can cause the system to respond incorrectly.

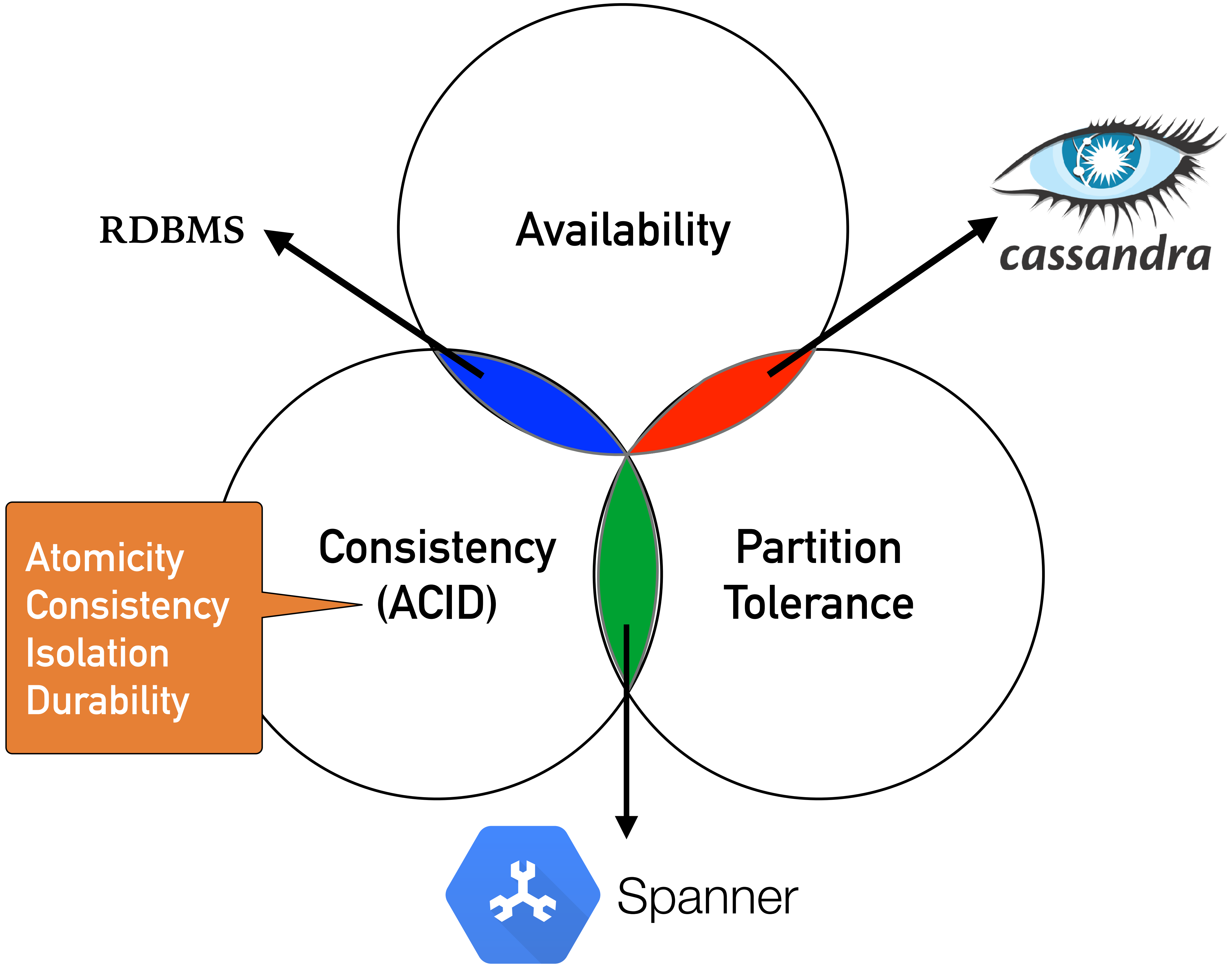
[E. Brewer]

Cassandra: Replication and Consistency



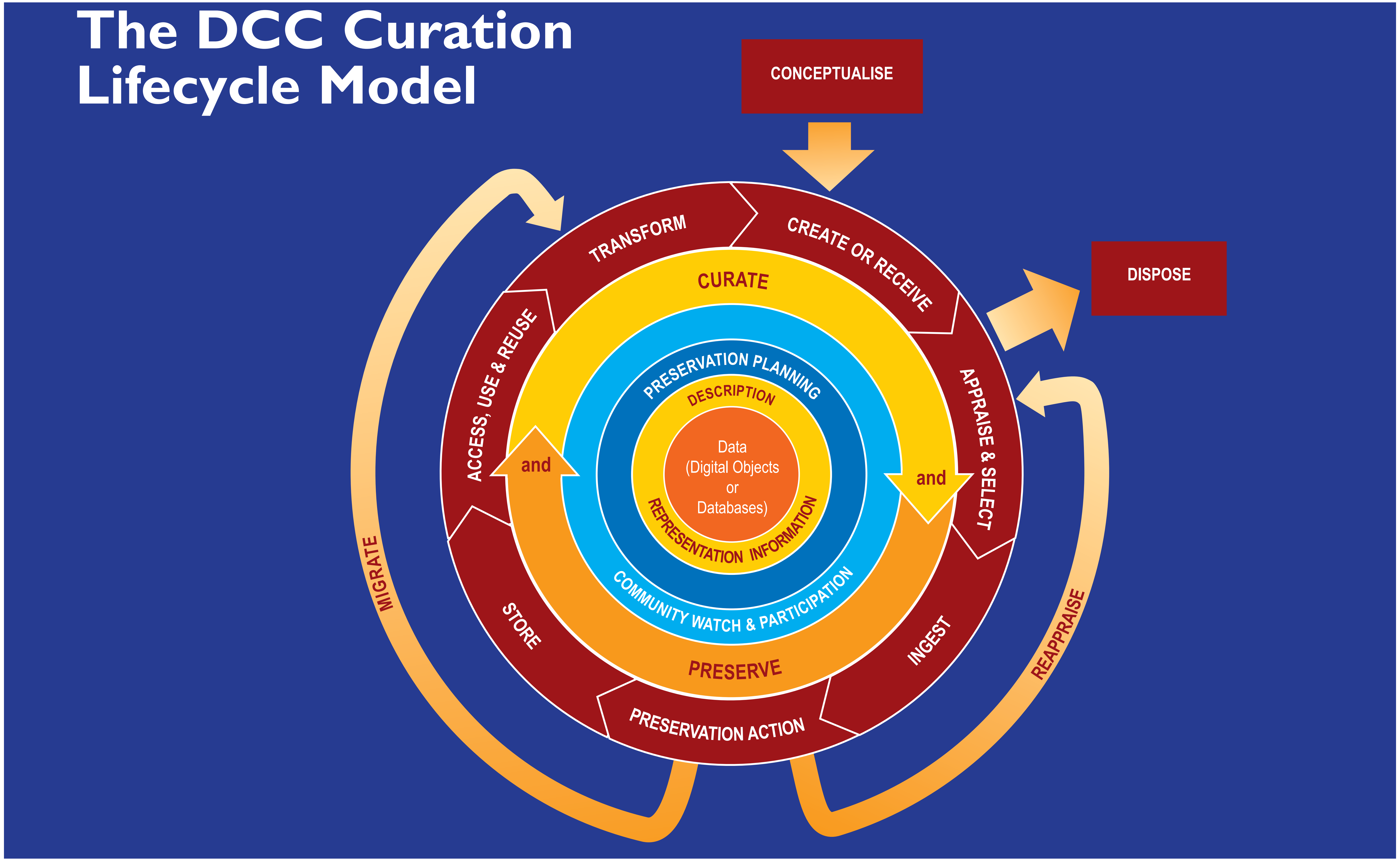
[R. Stupp]

Spanner: Google's NewSQL Cloud Database



- Which type of system is Spanner?
  - C: consistency, which implies a single value for shared data
  - A: 100% availability, for both reads and updates
  - P: tolerance to network partitions
- Which two?
  - CA: close, but not totally available
  - So actually **CP**

Data Curation



[DCC]

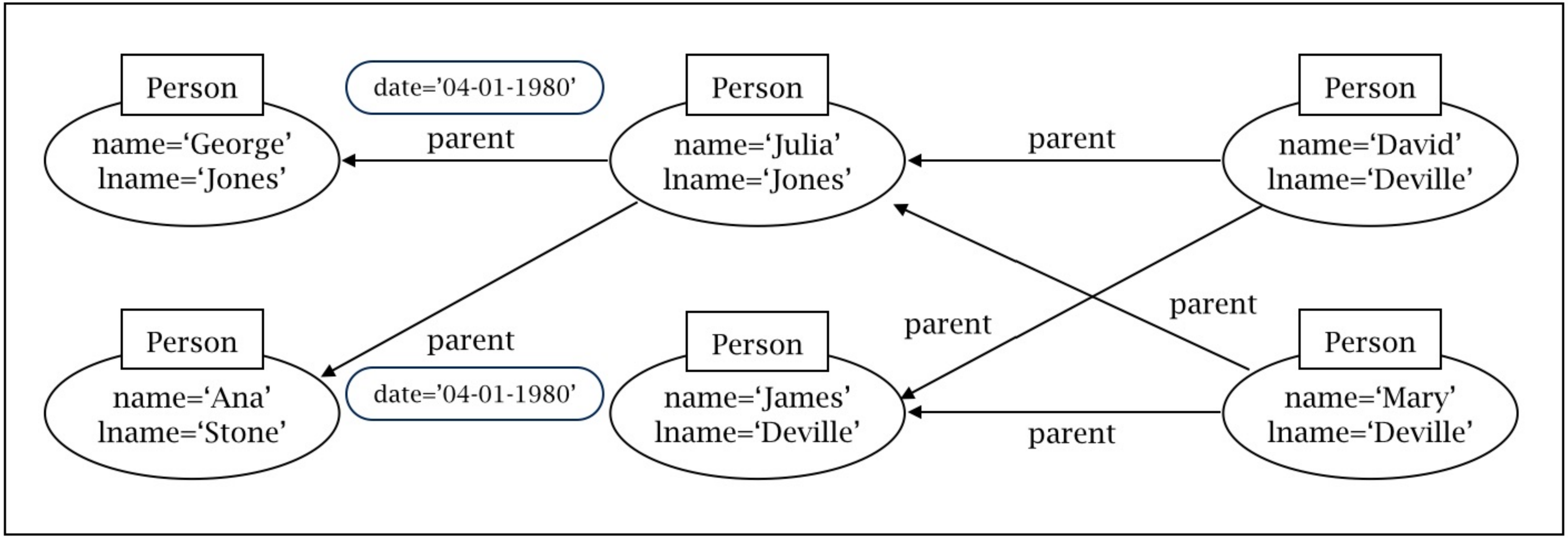
FAIR Principles

- Findable: Metadata and data should be easy to find for both humans and computers
- Accessible: Users need to know how data can be accessed, possibly including authentication and authorization
- Interoperable: Can be integrated with other data, and can interoperate with applications or workflows for analysis, storage, and processing
- Reusable: Optimize the reuse of data. Metadata and data should be well-described so they can be replicated and/or combined in different settings

[GO FAIR]

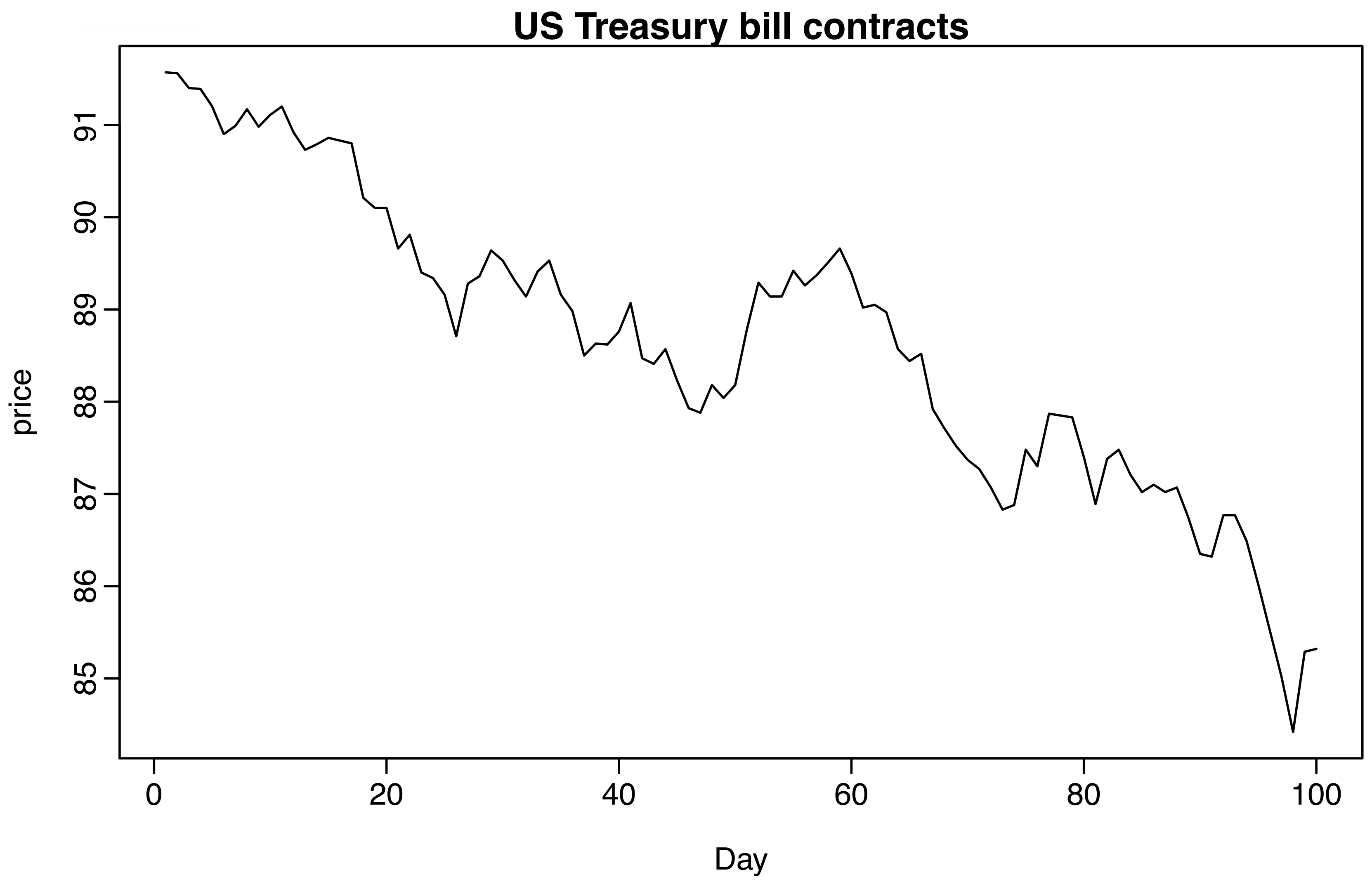
Graph Databases focus on relationships

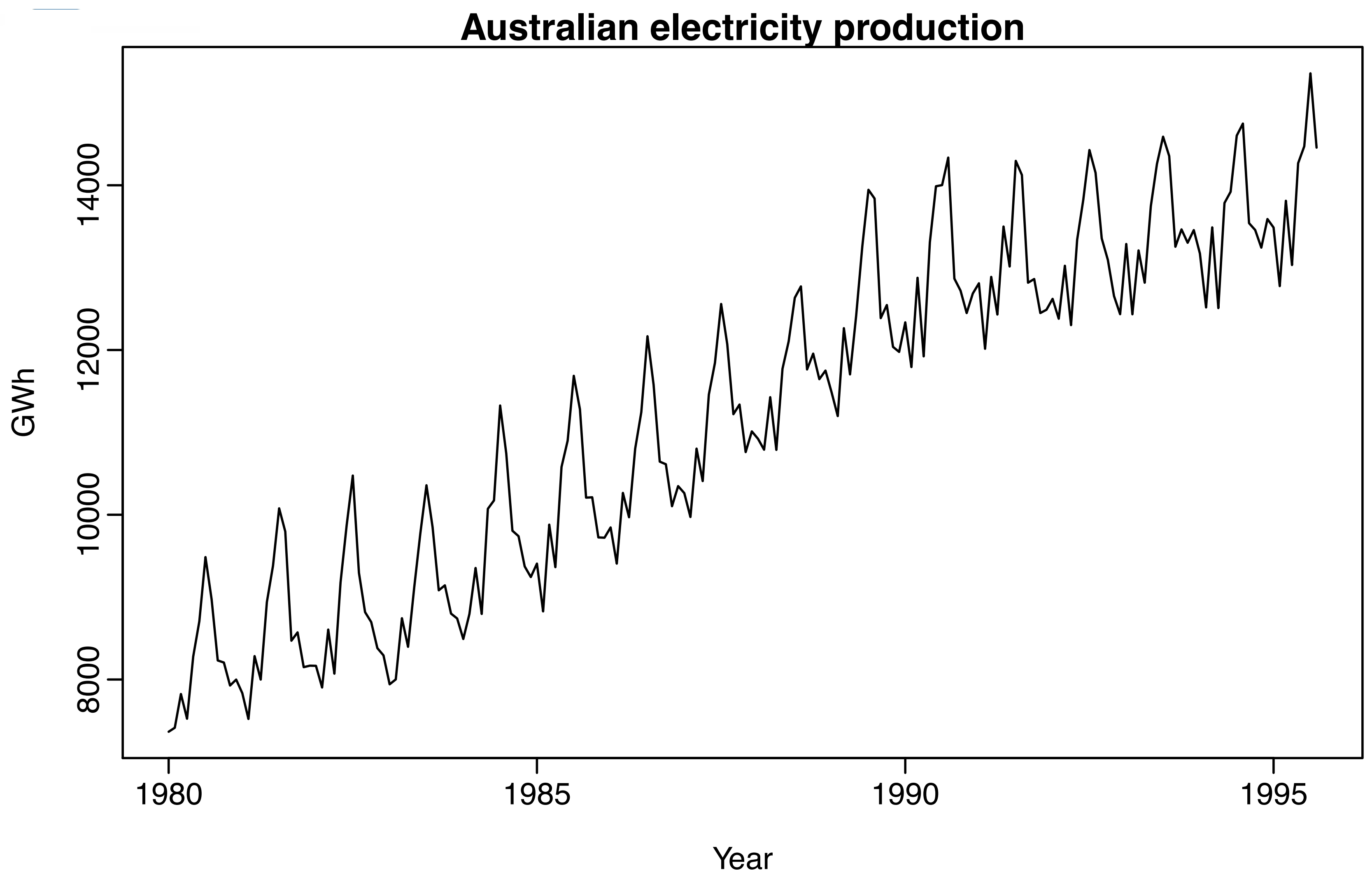
- Directed, labelled, attributed multigraph
- Properties are **key/value pairs** that represent metadata for nodes and edges

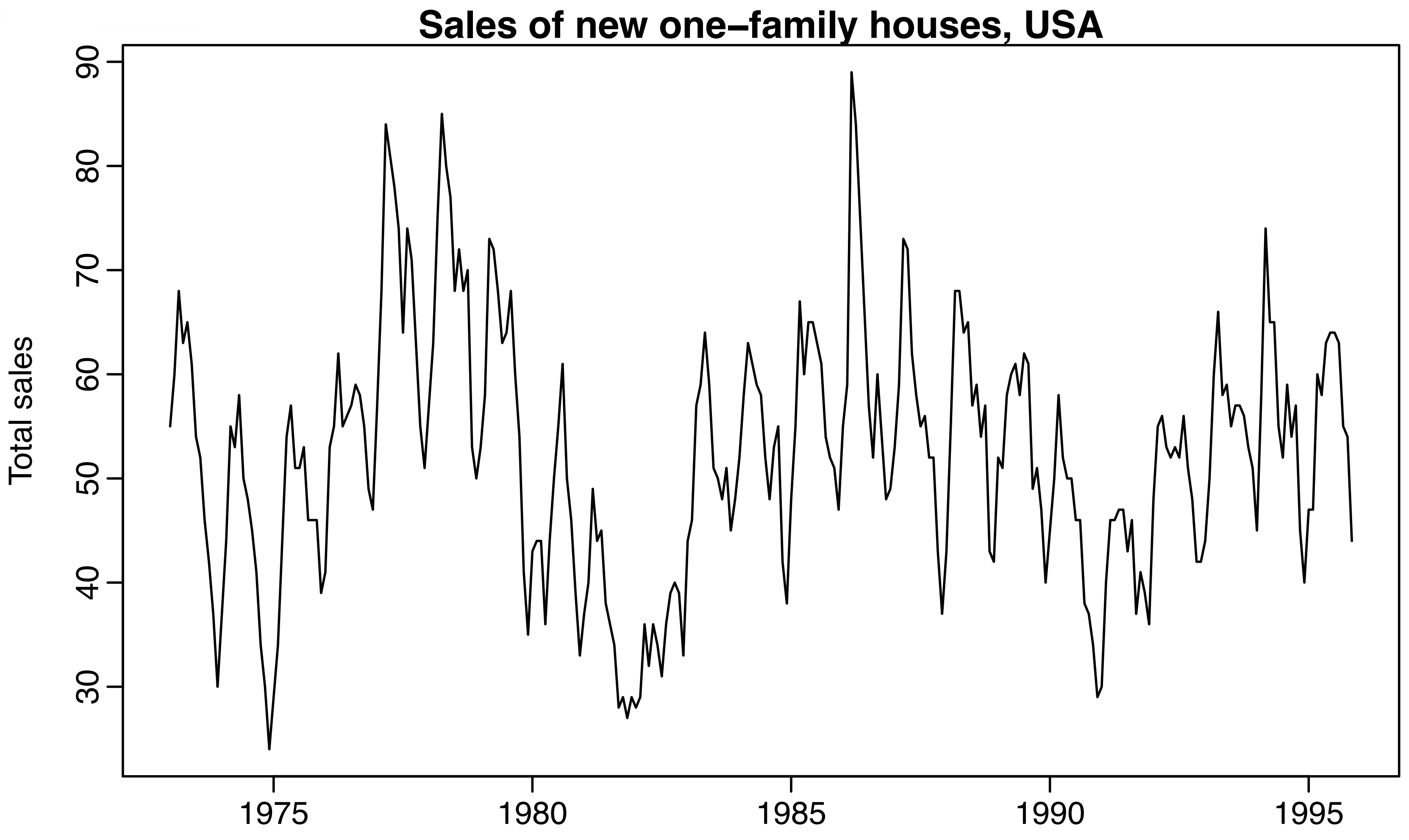


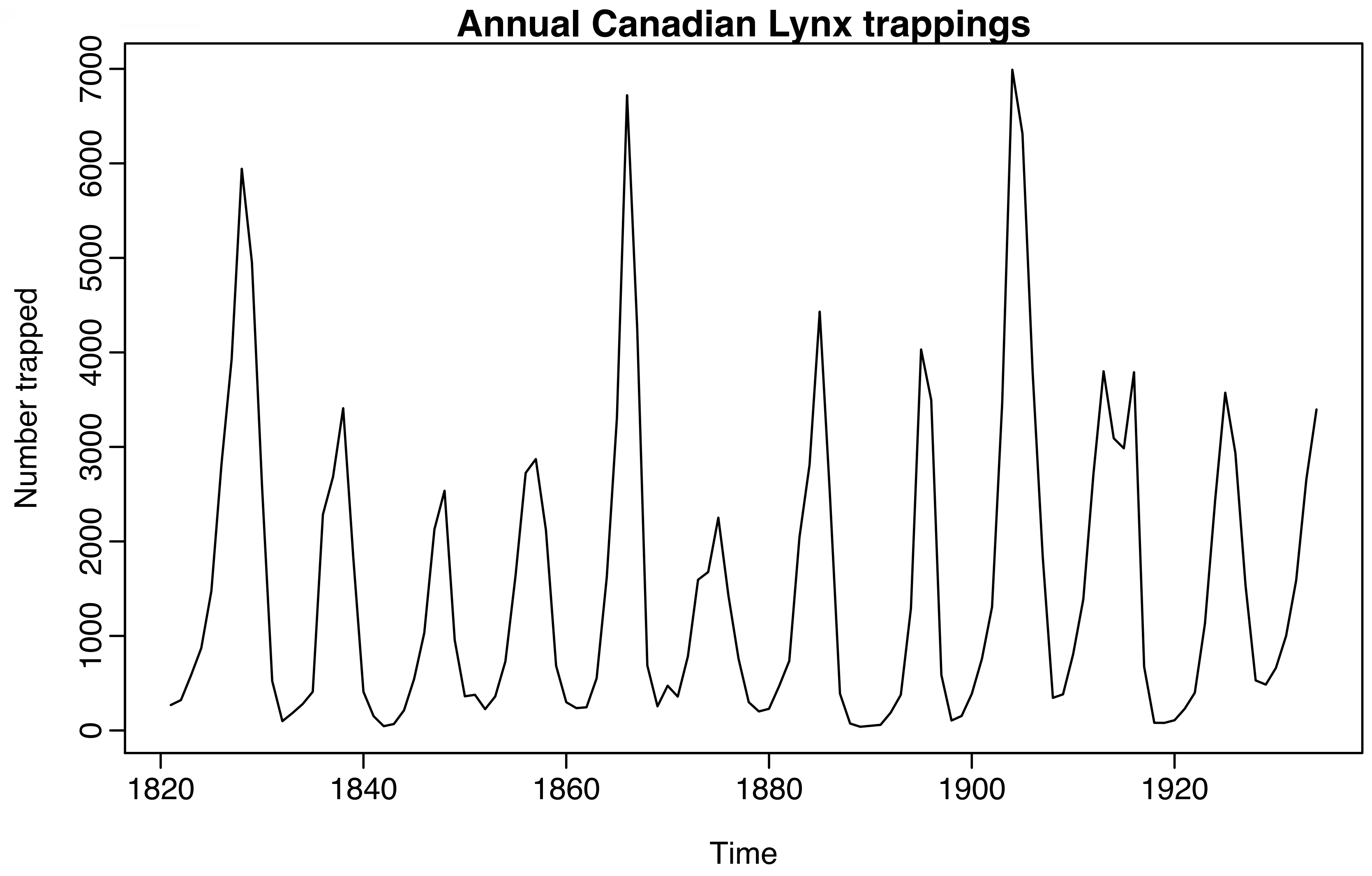
[R. Angles and C. Gutierrez, 2017]

Time Series Data





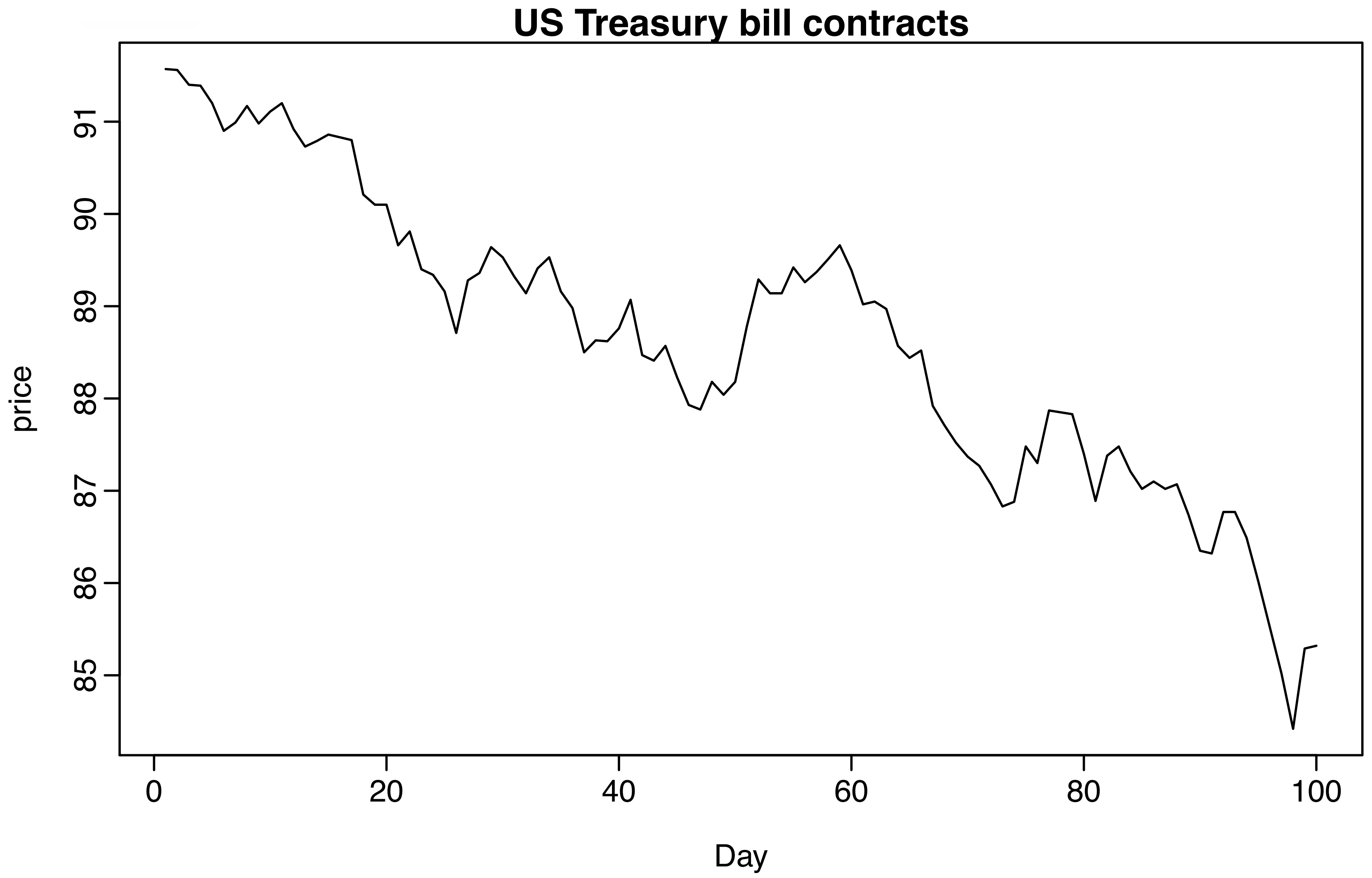


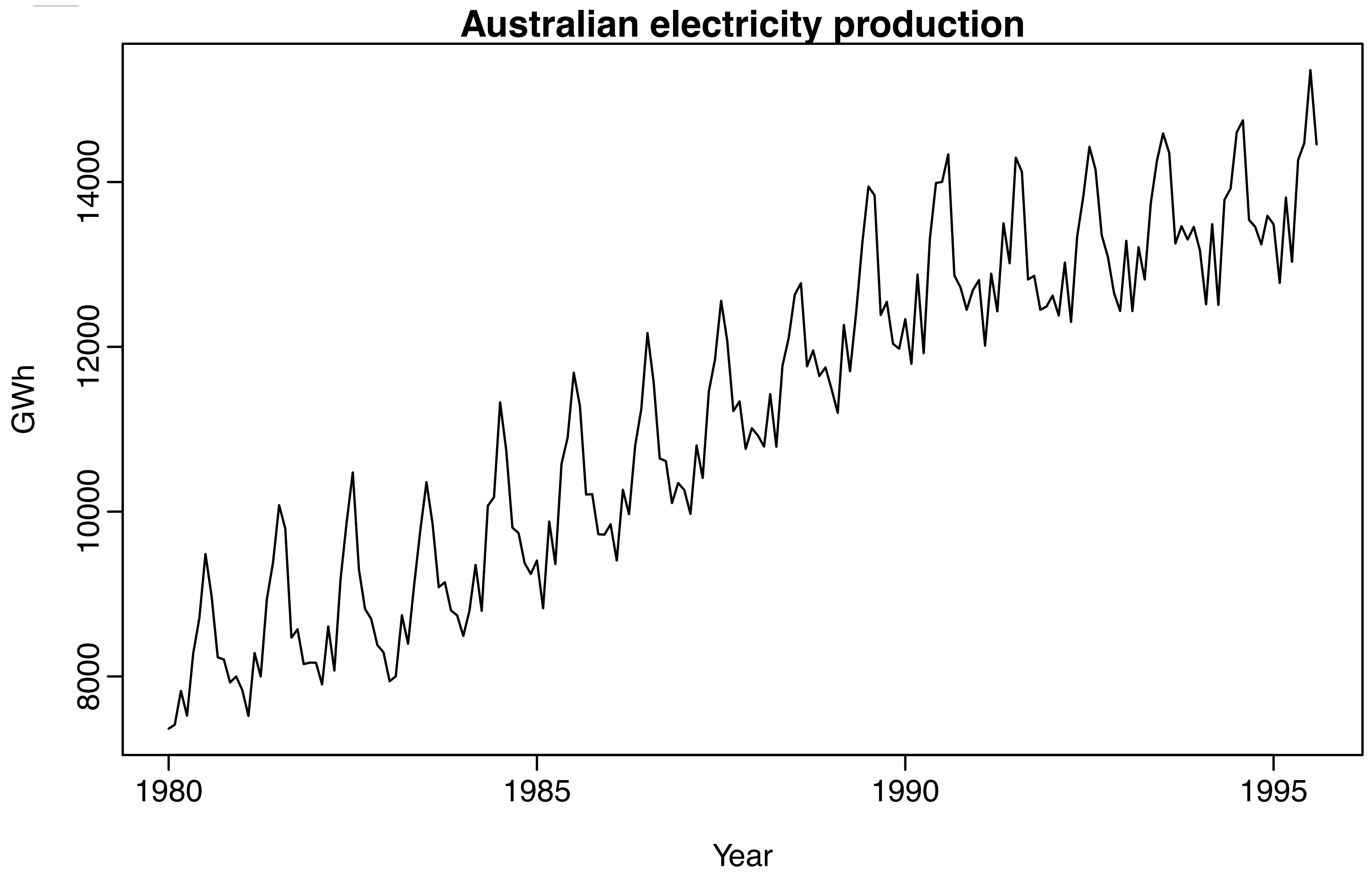


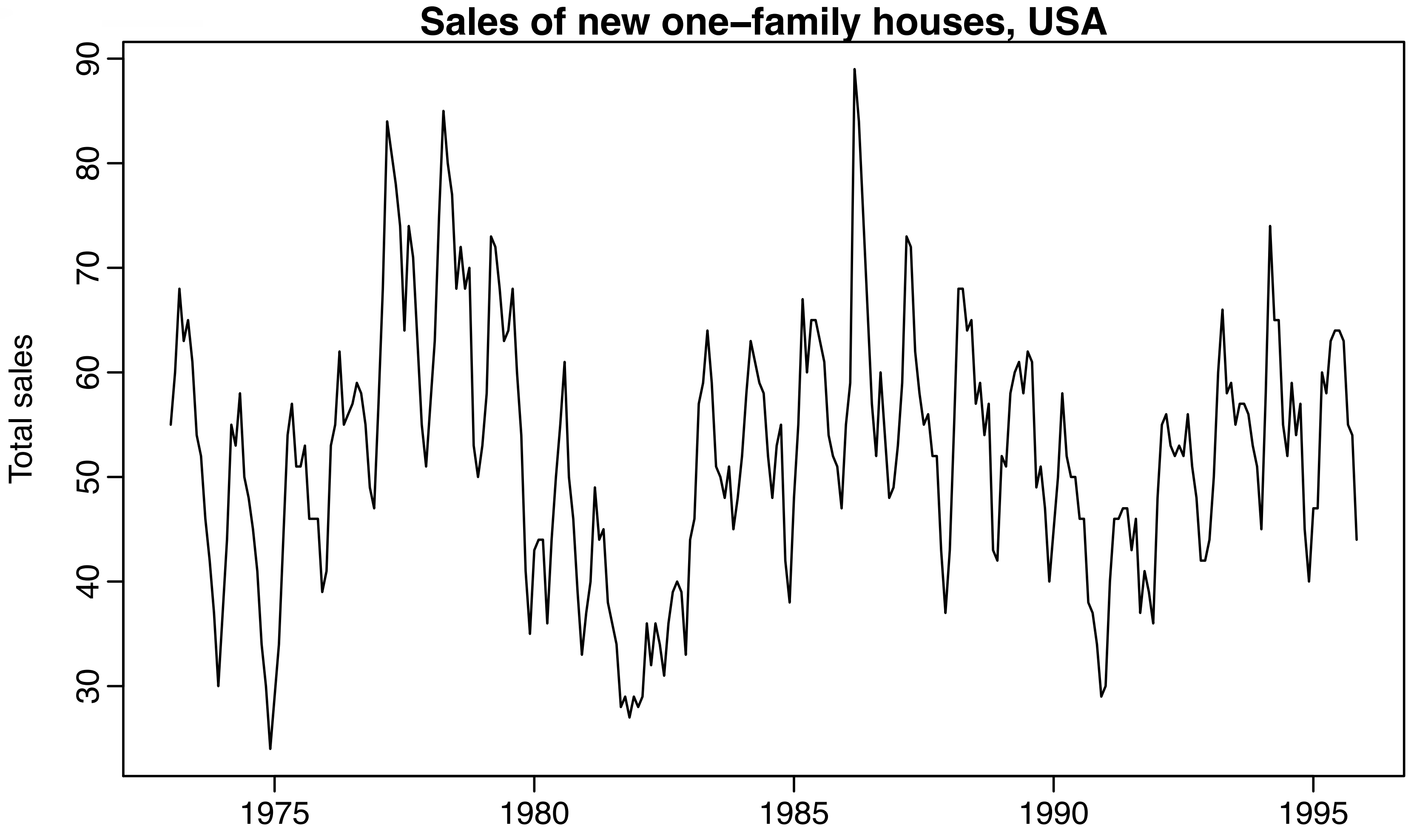
[R. J. Hyndman]

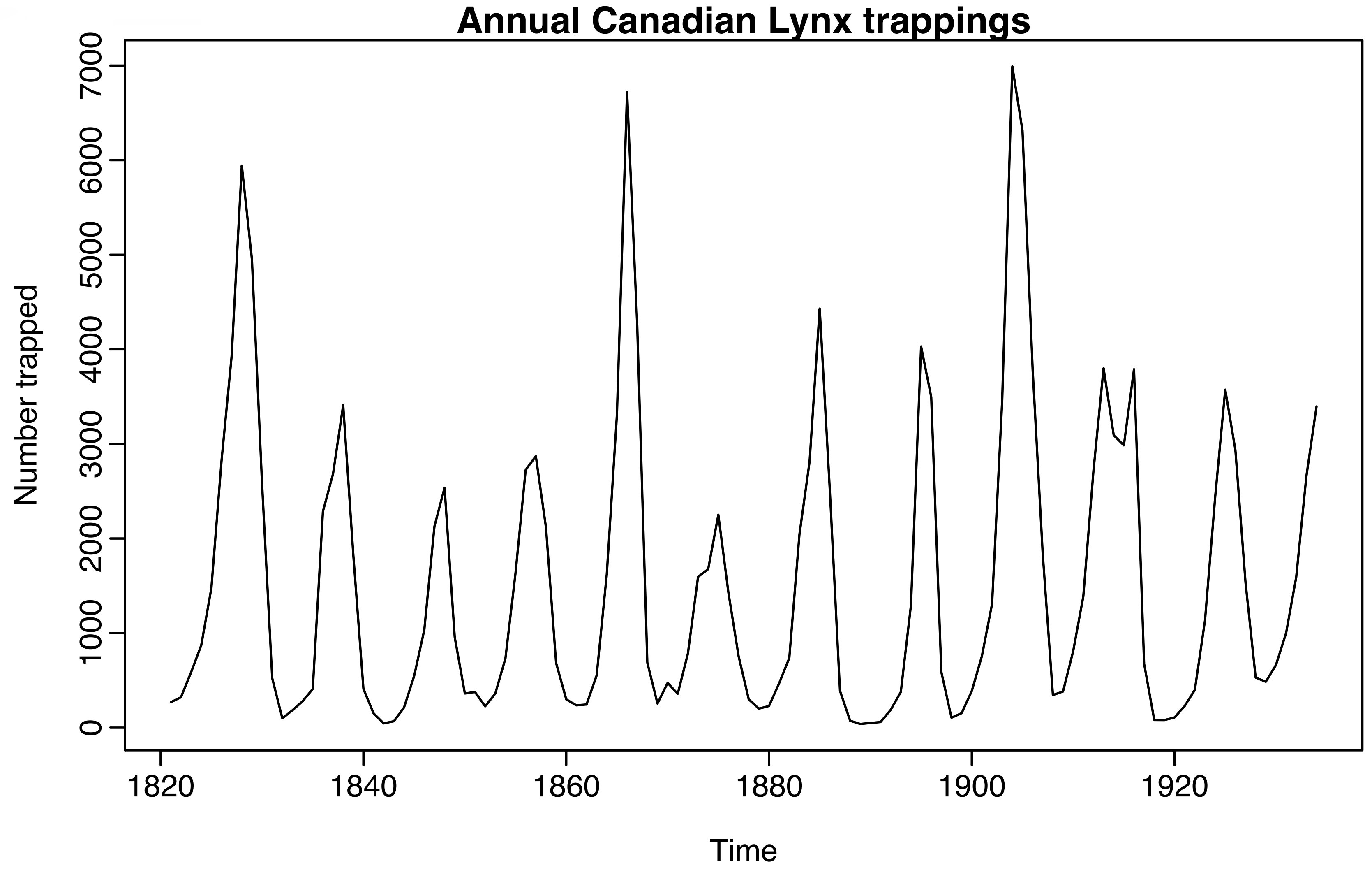
Time Series Data

Trend


US Treasury bill contracts


Australian electricity production

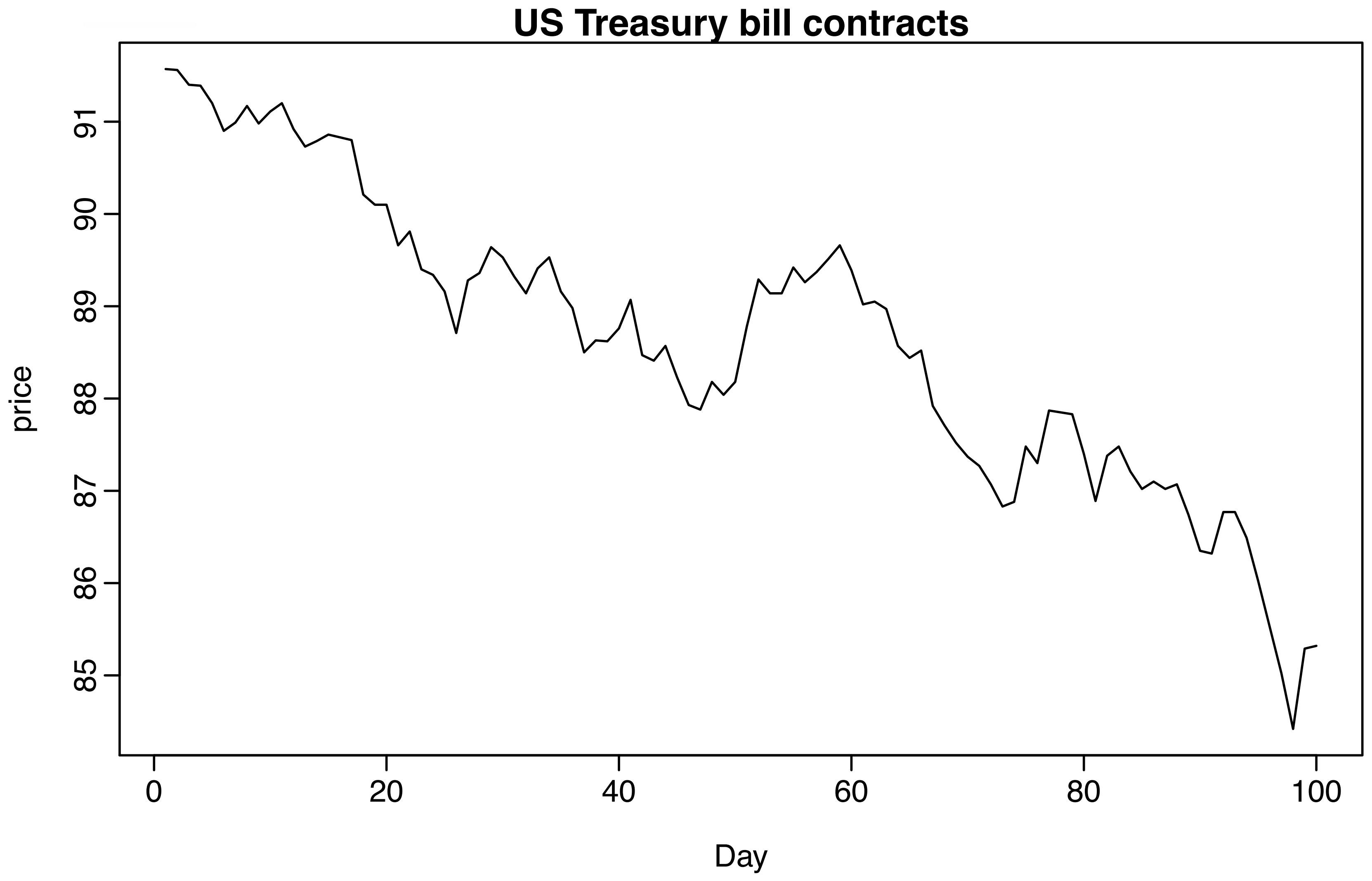

Sales of new one–family houses, USA

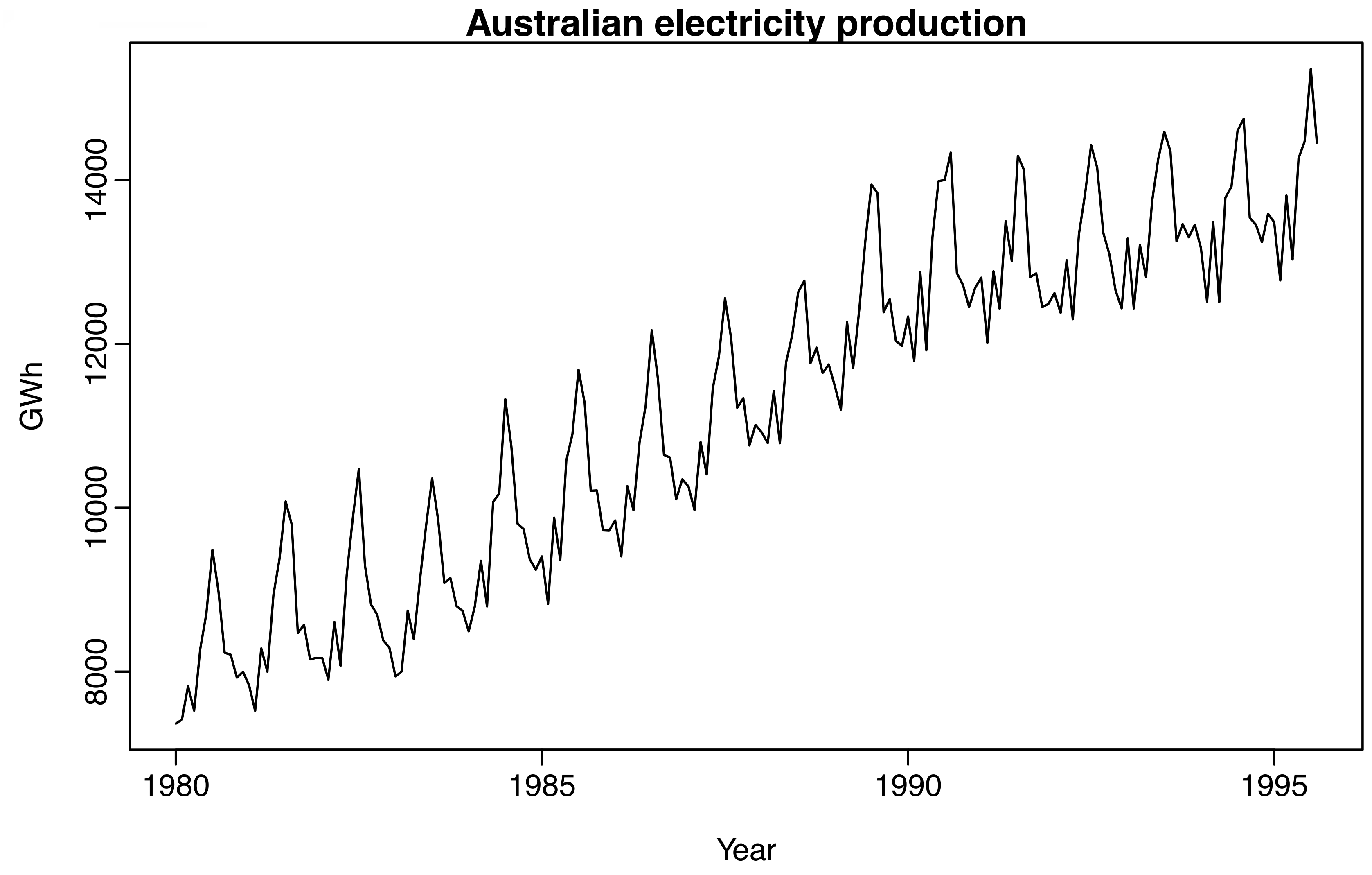

Annual Canadian Lynx trappings

[R. J. Hyndman]

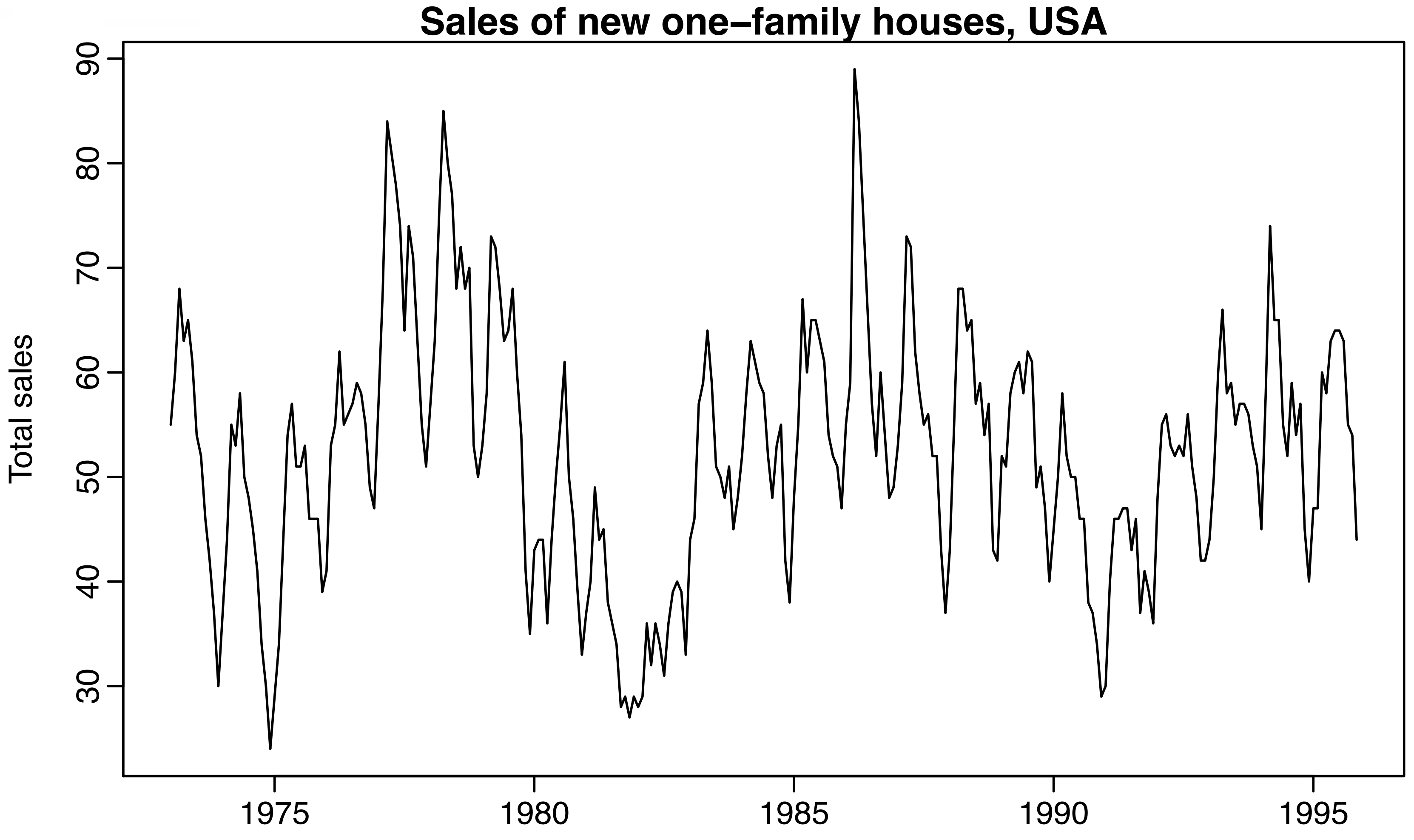
Time Series Data

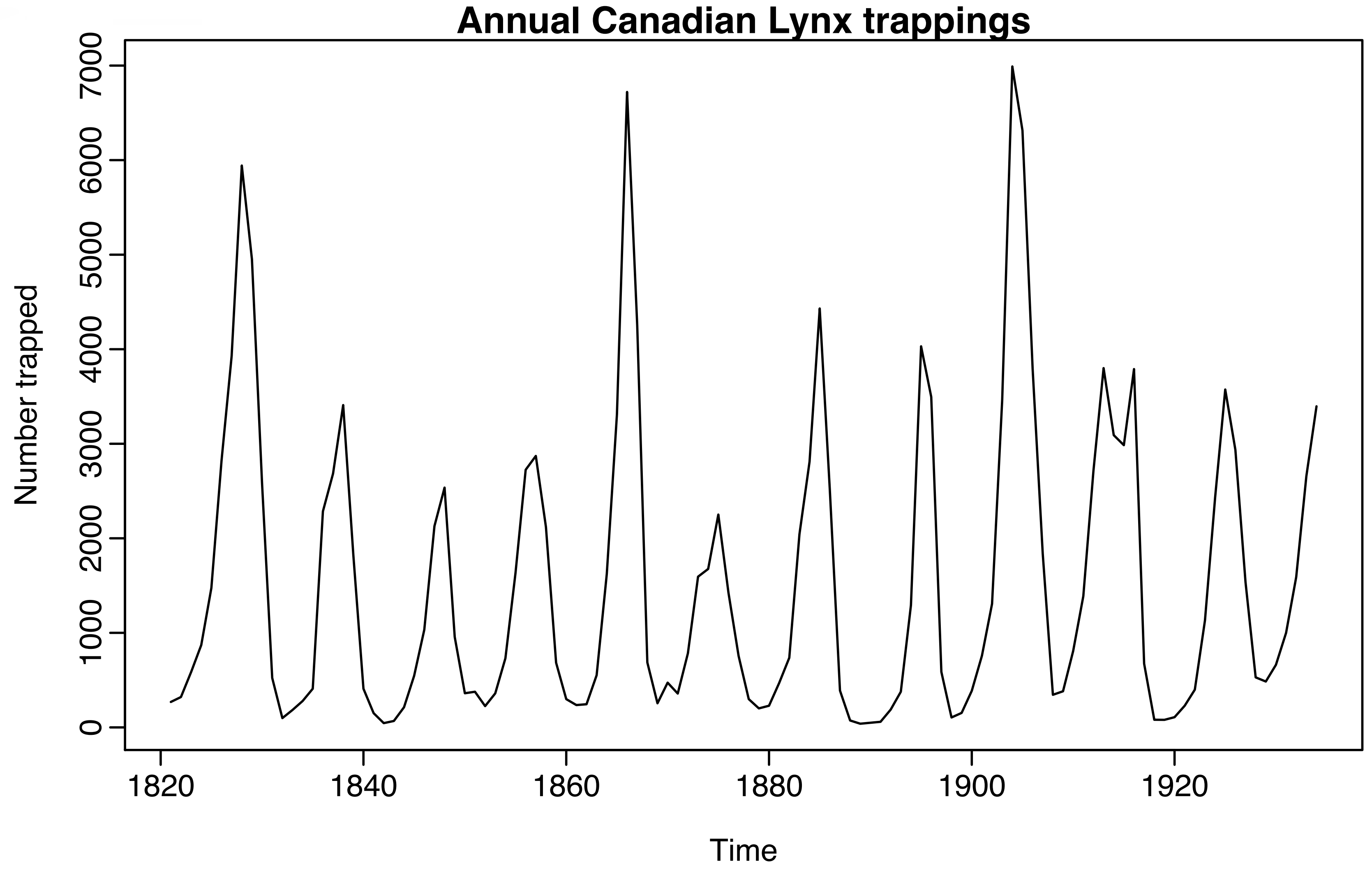
Trend





Trend + Seasonality

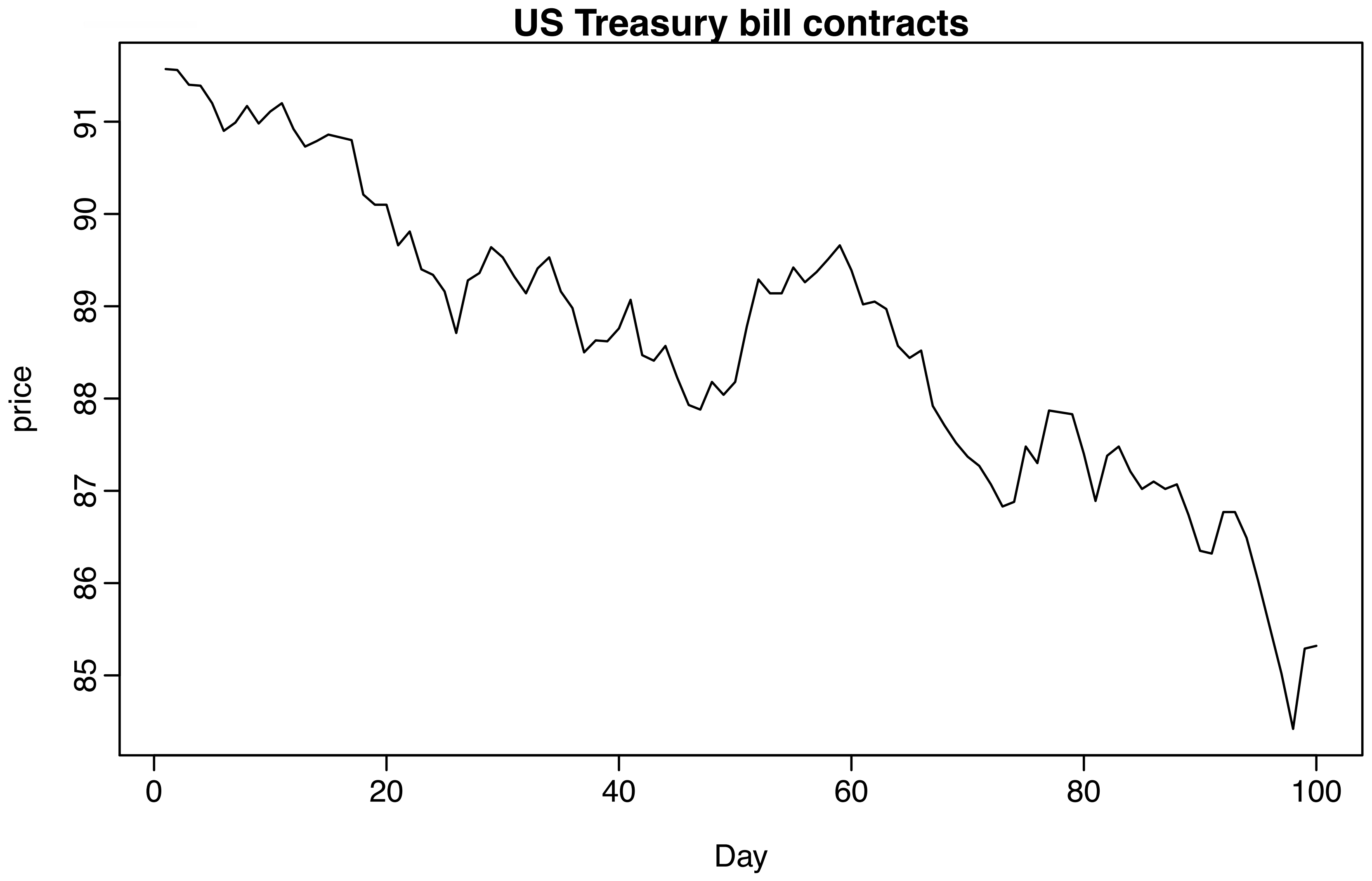


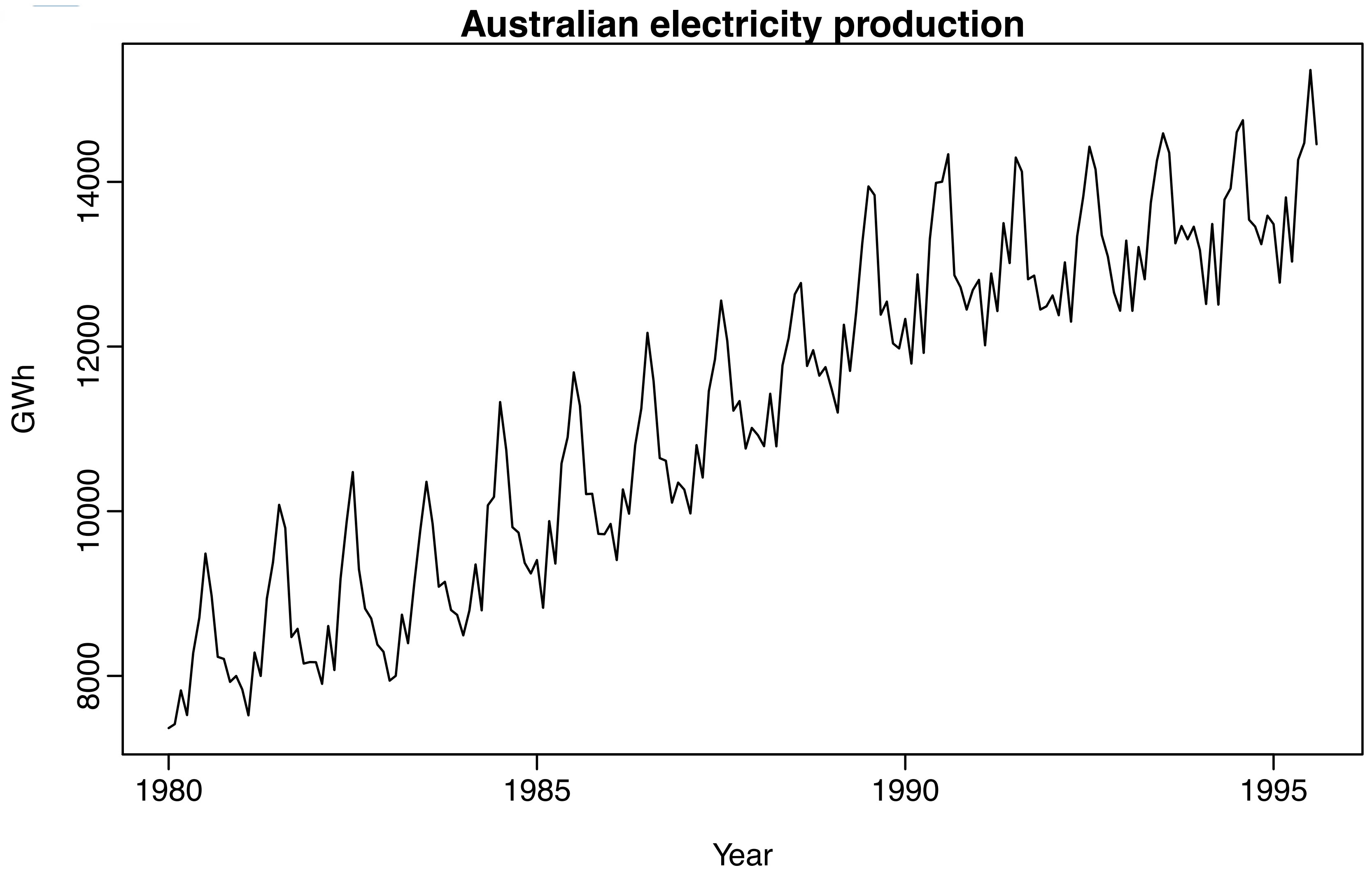


[R. J. Hyndman]

Time Series Data

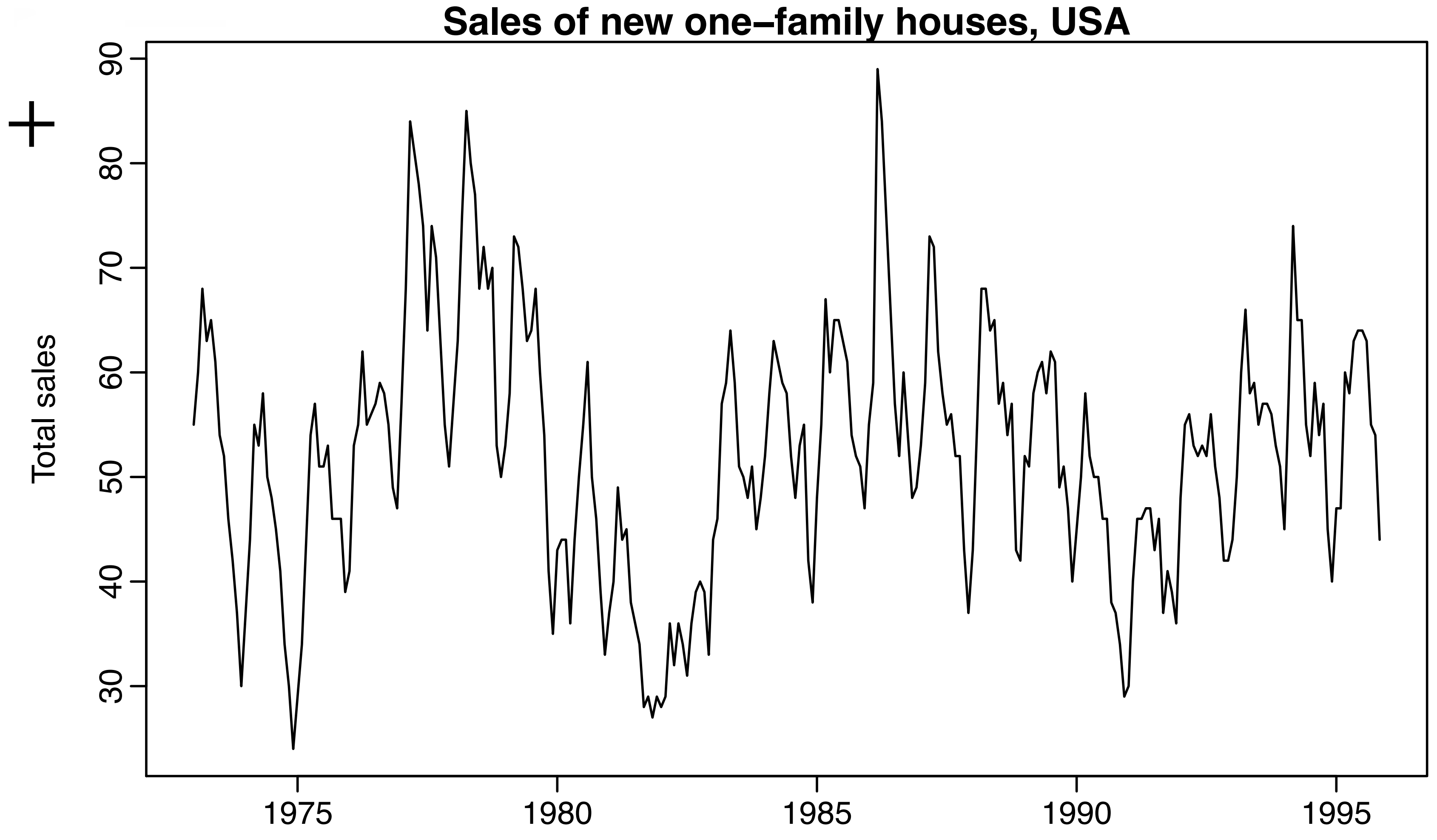
Trend

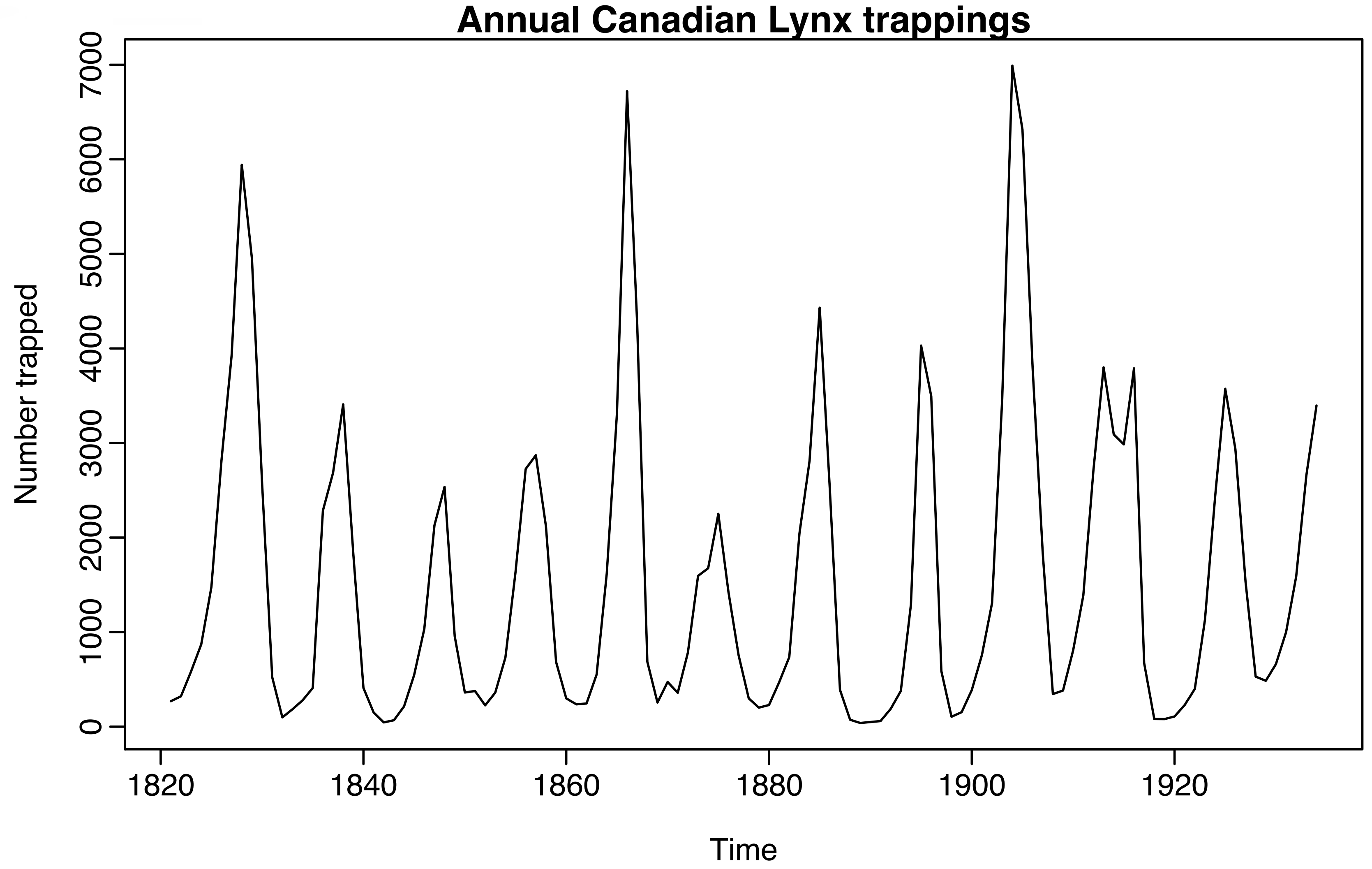

US Treasury bill contracts


Australian electricity production

Trend + Seasonality

Seasonality + Cyclic

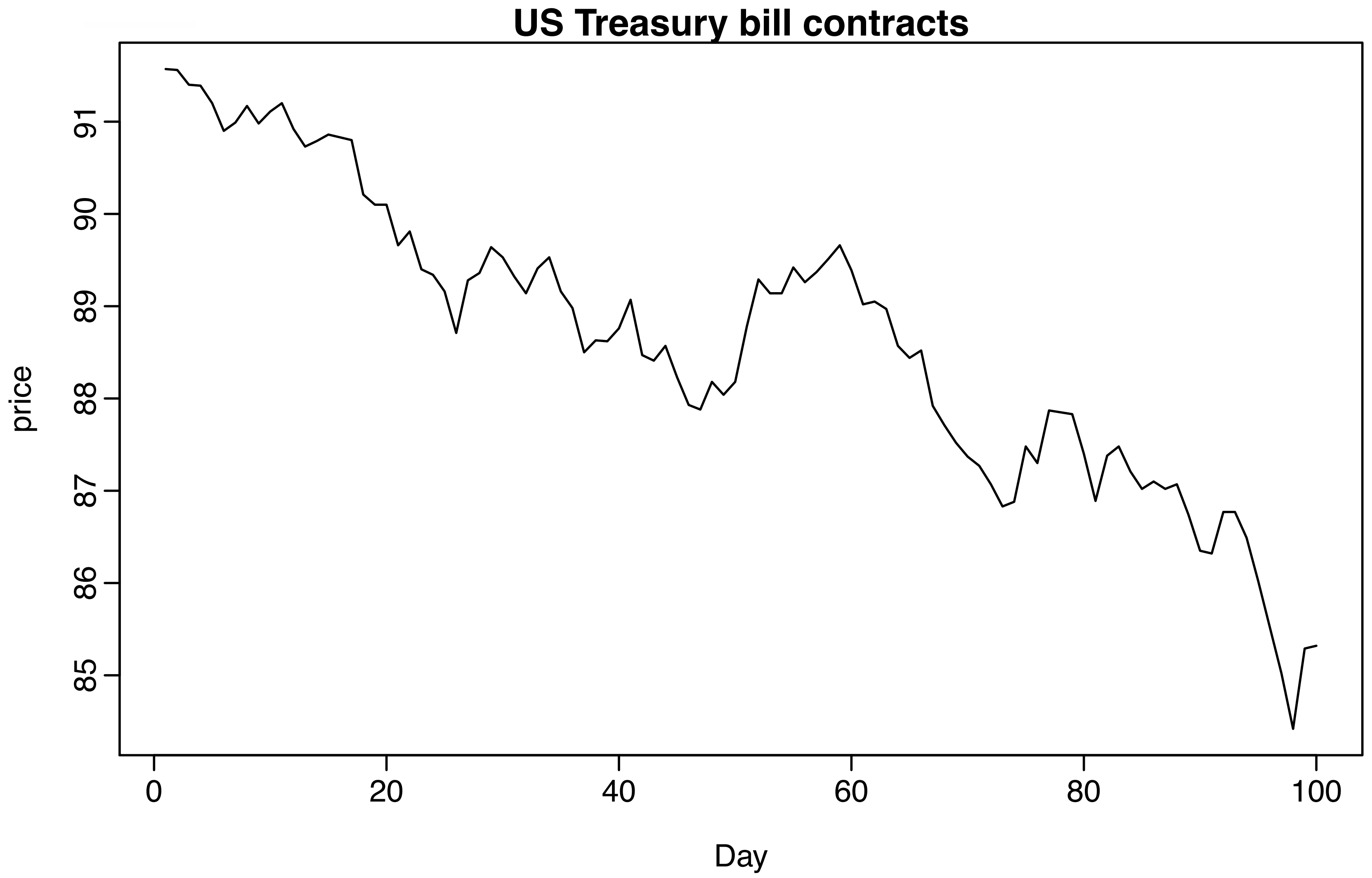

Sales of new one–family houses, USA

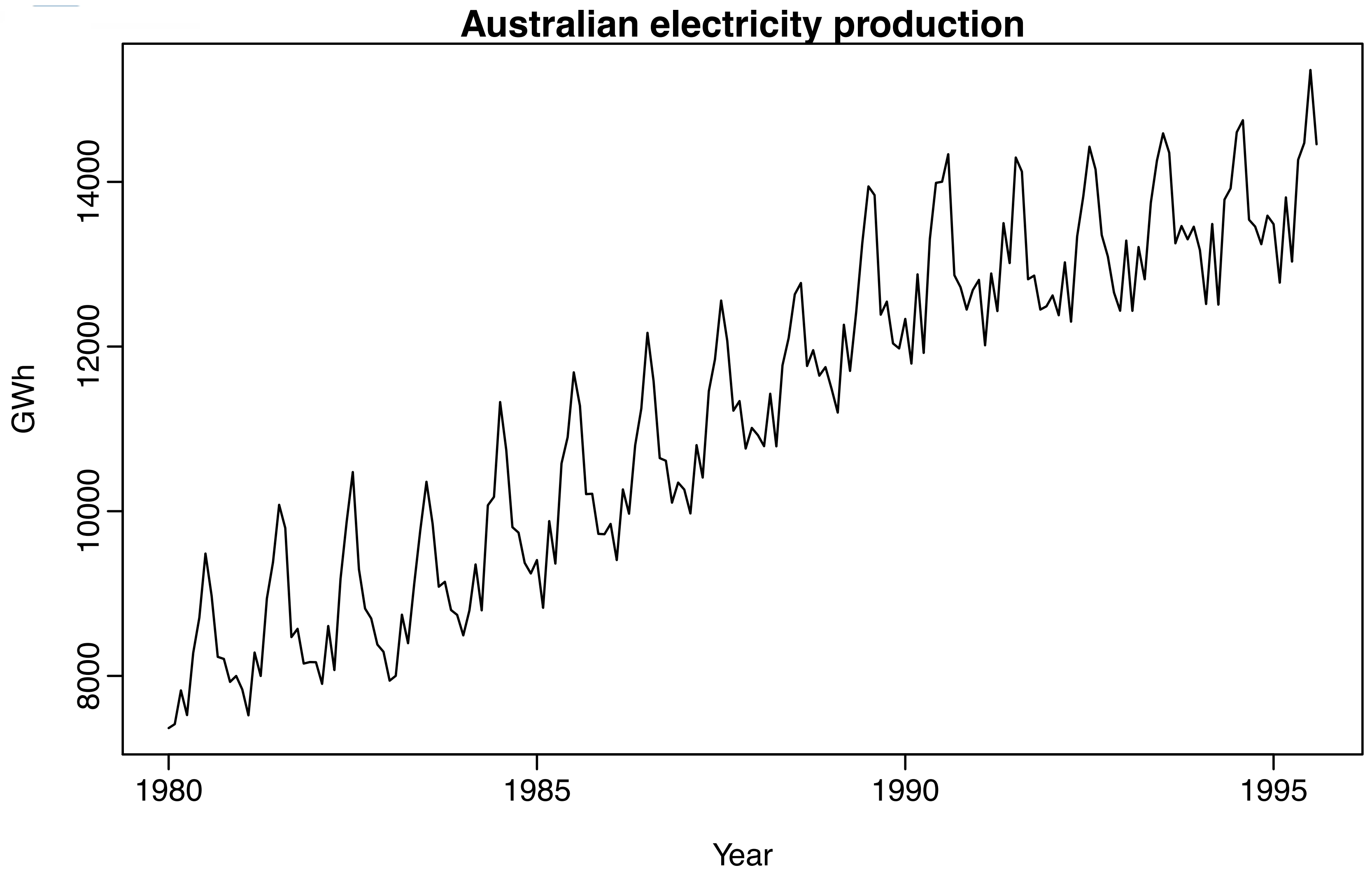

Annual Canadian Lynx trappings

[R. J. Hyndman]

Time Series Data

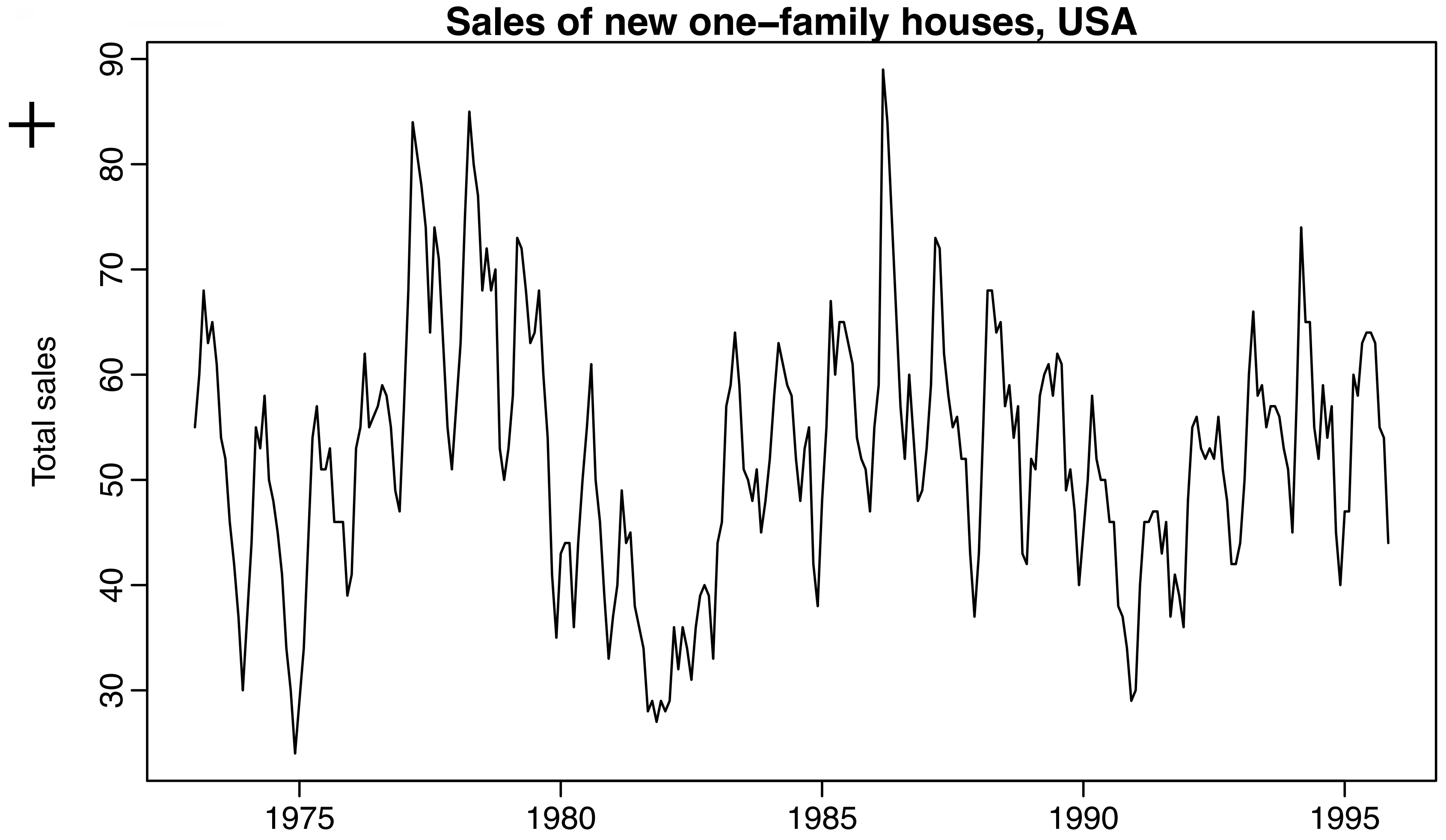
Trend

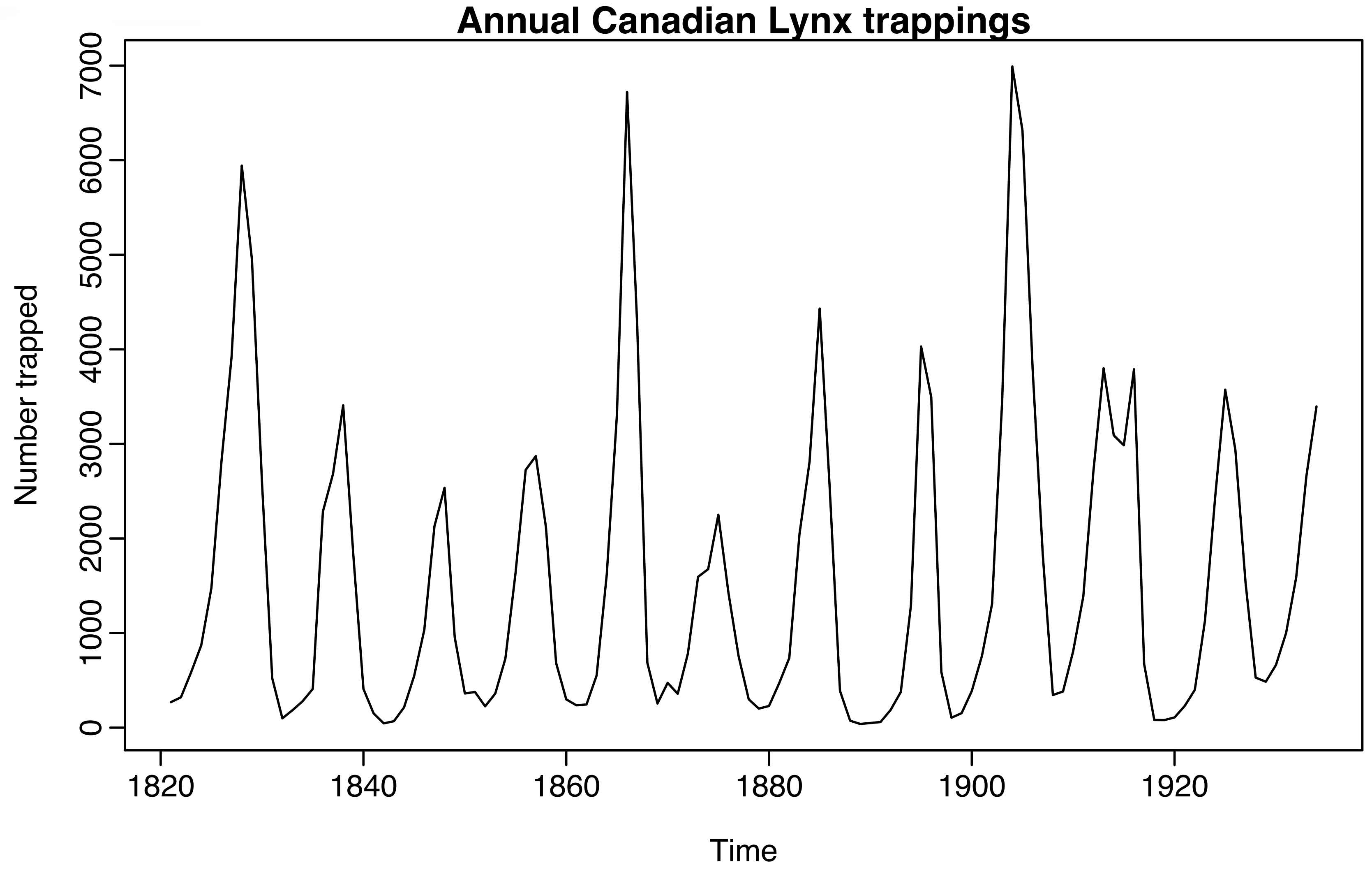




Trend + Seasonality

Seasonality + Cyclic

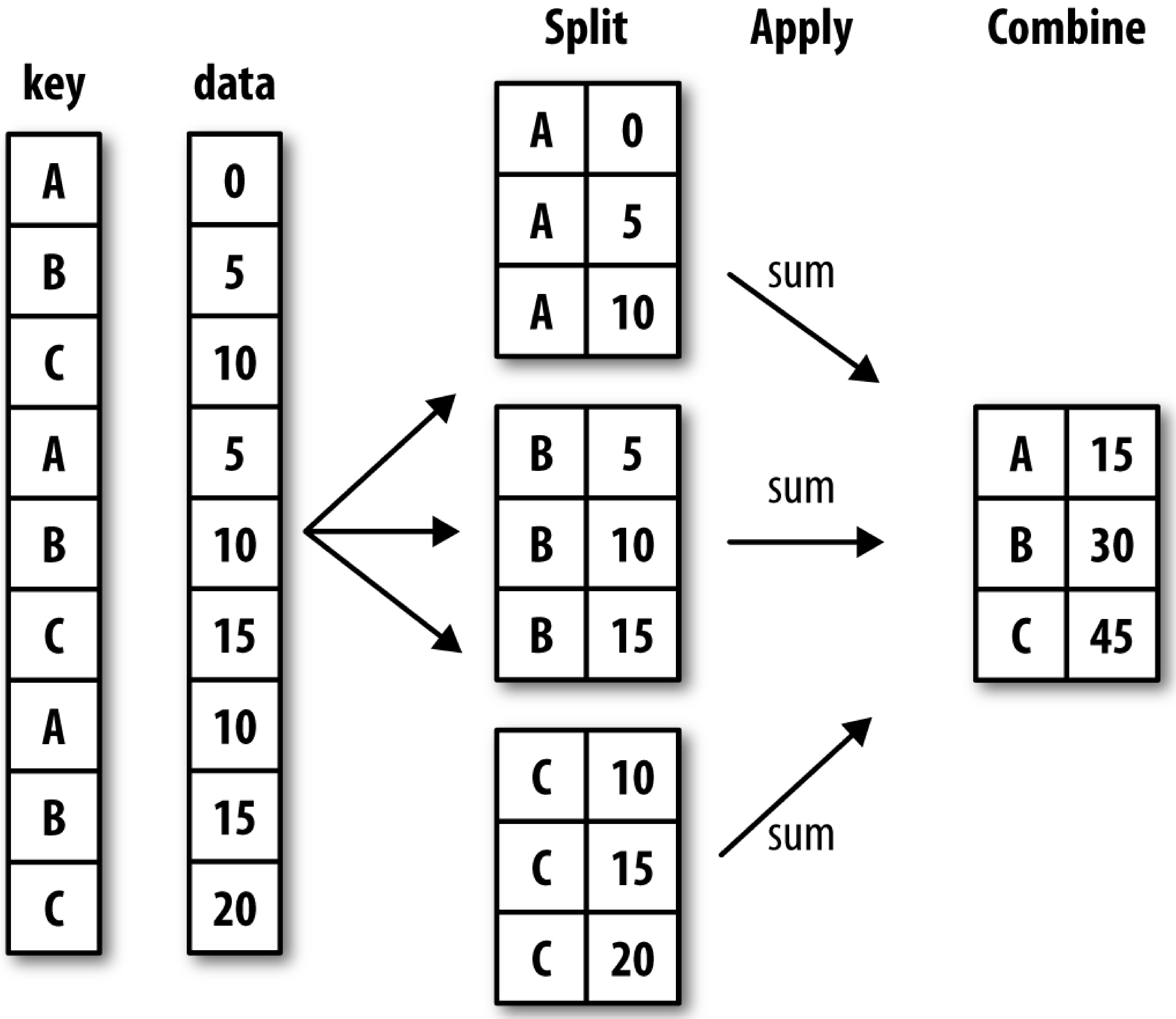




Stationary

[R. J. Hyndman]

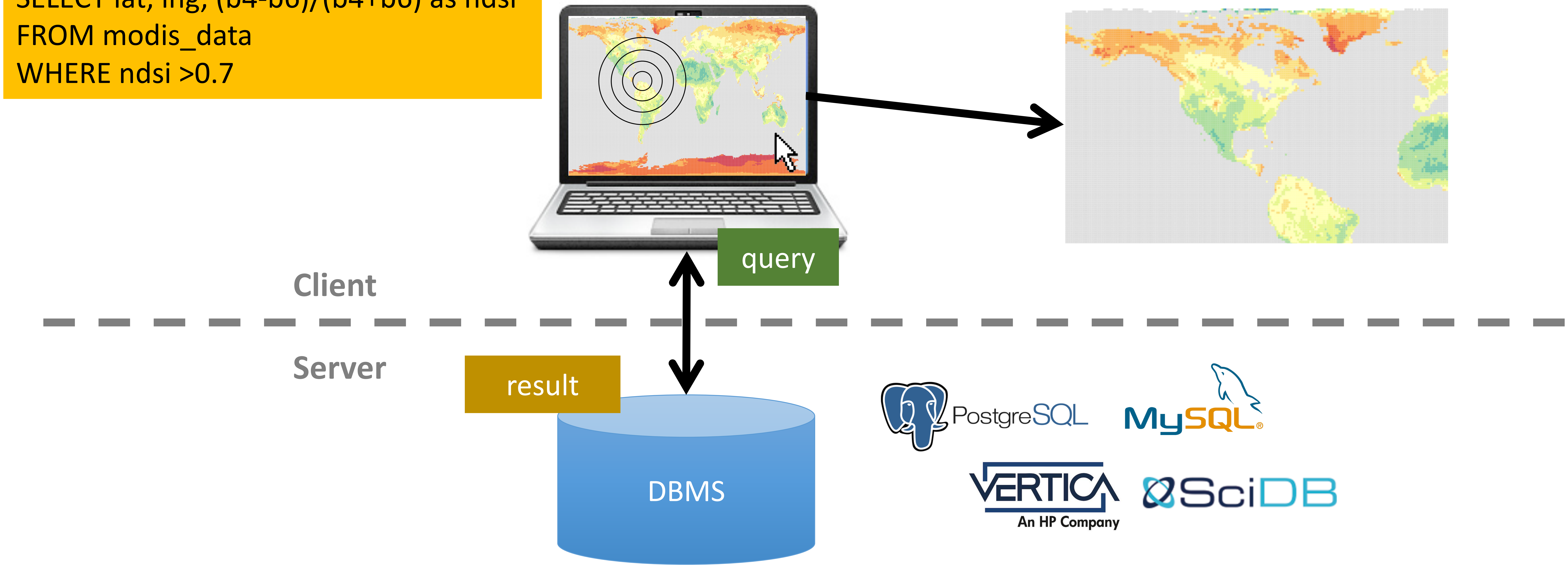
Split-Apply-Combine



[W. McKinney, Python for Data Analysis]

Interactive Exploration of Spatial Data

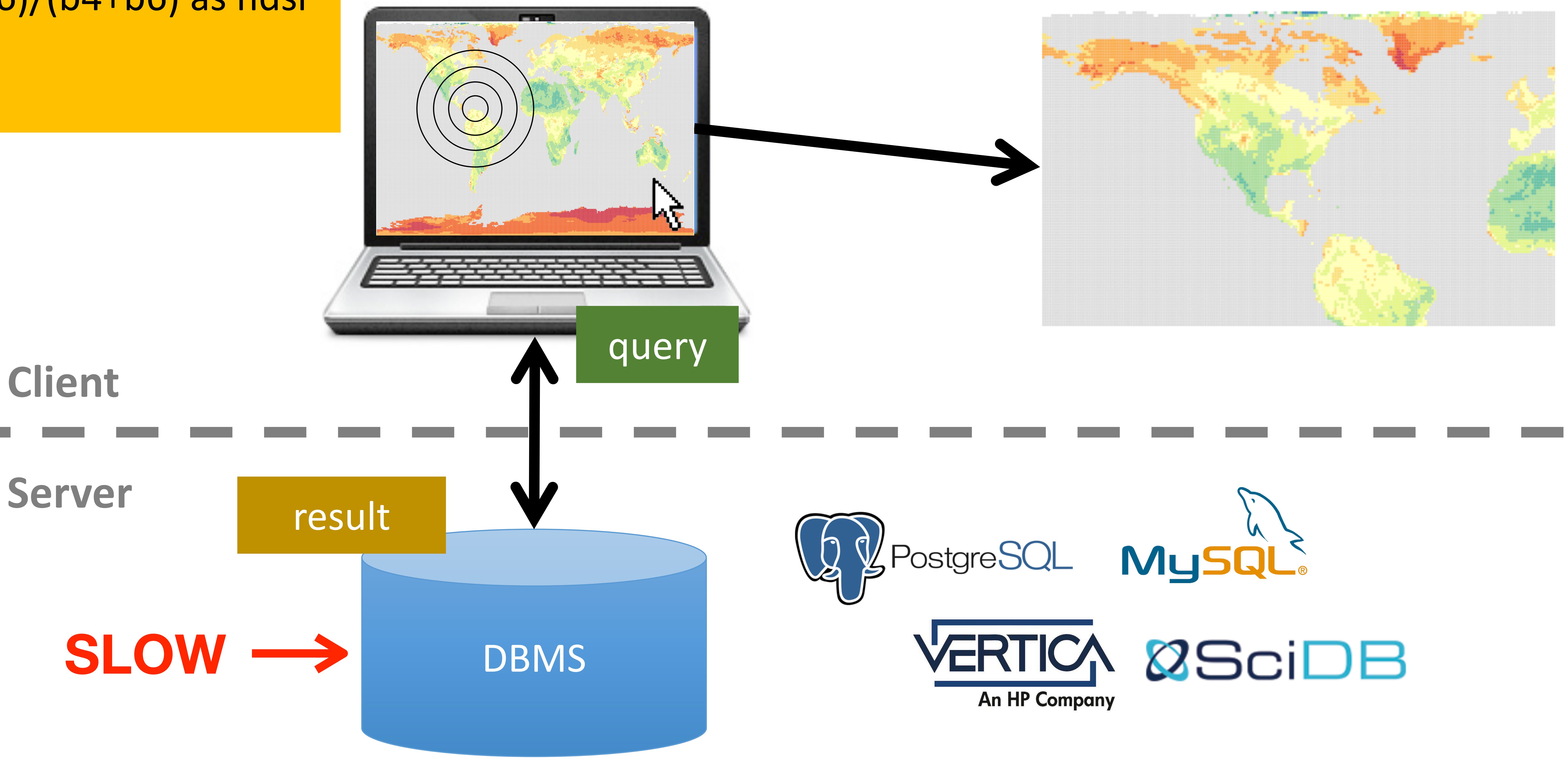
```
SELECT lat, lng, (b4-b6)/(b4+b6) as ndsi
FROM modis_data
WHERE ndsi >0.7
```



[L. Battle, 2017]

Interactive Exploration of Spatial Data

```
SELECT lat, lng, (b4-b6)/(b4+b6) as ndsi
FROM modis_data
WHERE ndsi >0.7
```



query

Client

Server

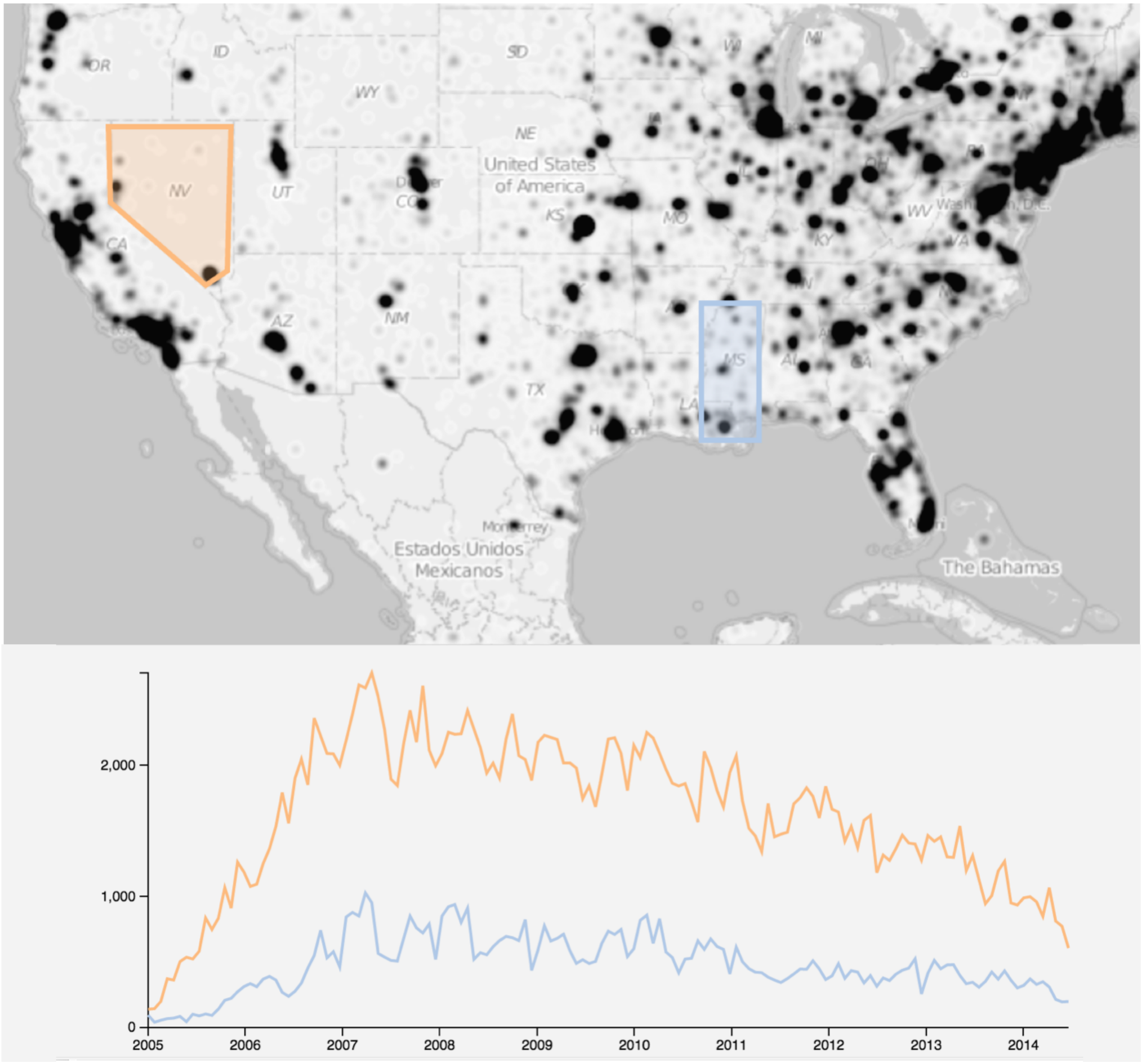
result

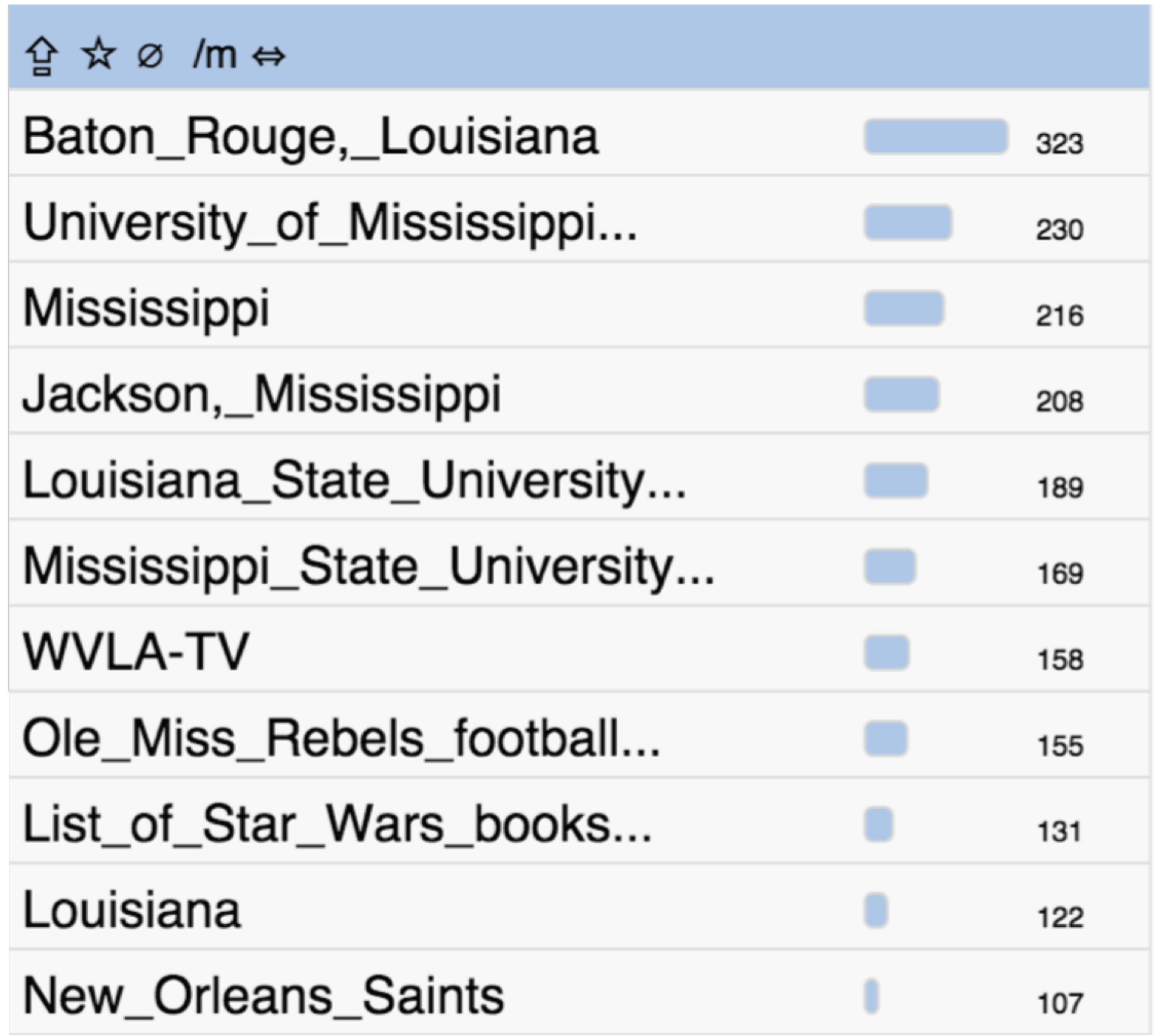
SLOW →

DBMS

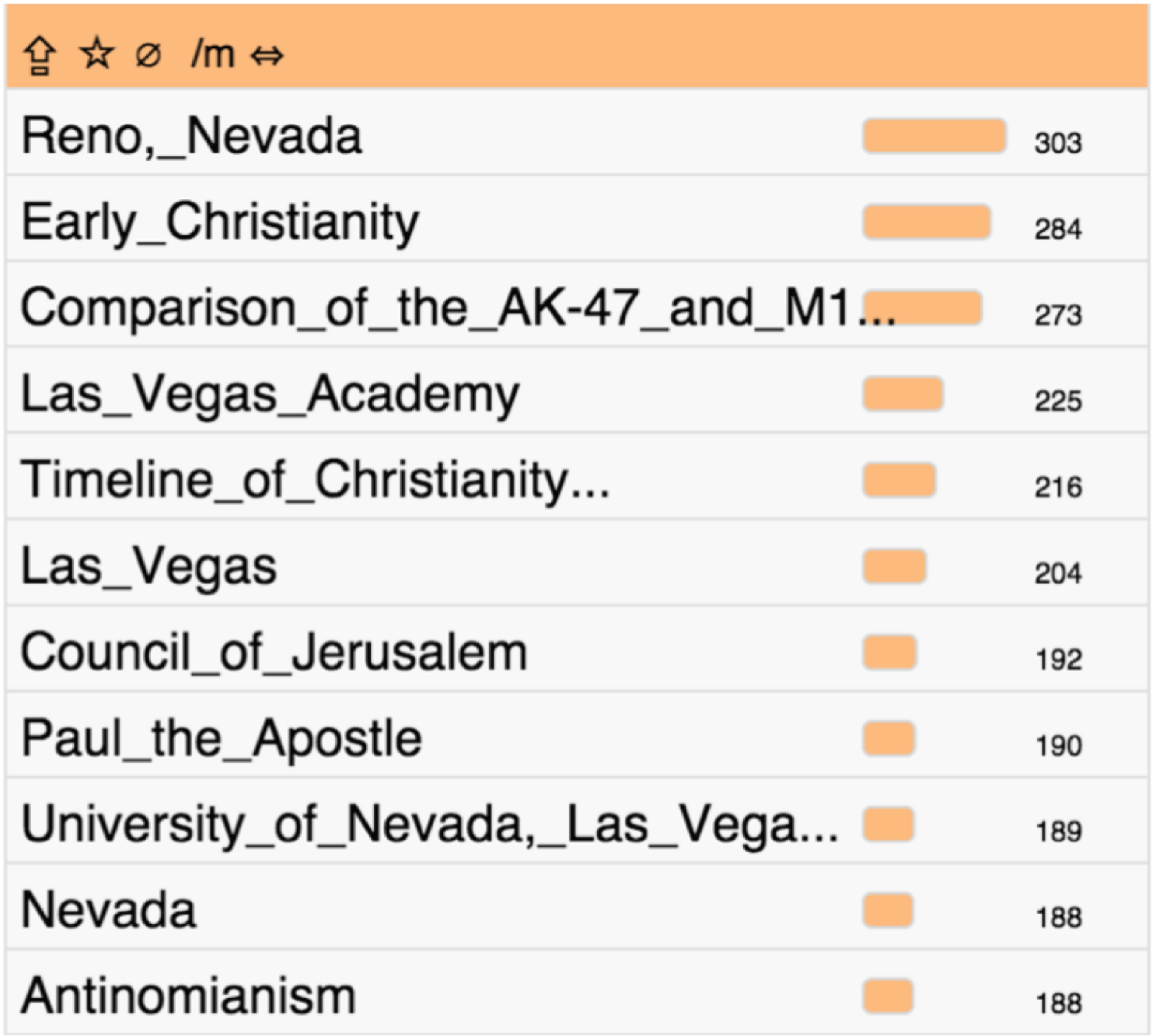
[L. Battle, 2017]

Spatial Data: NanoCubes and TopKube





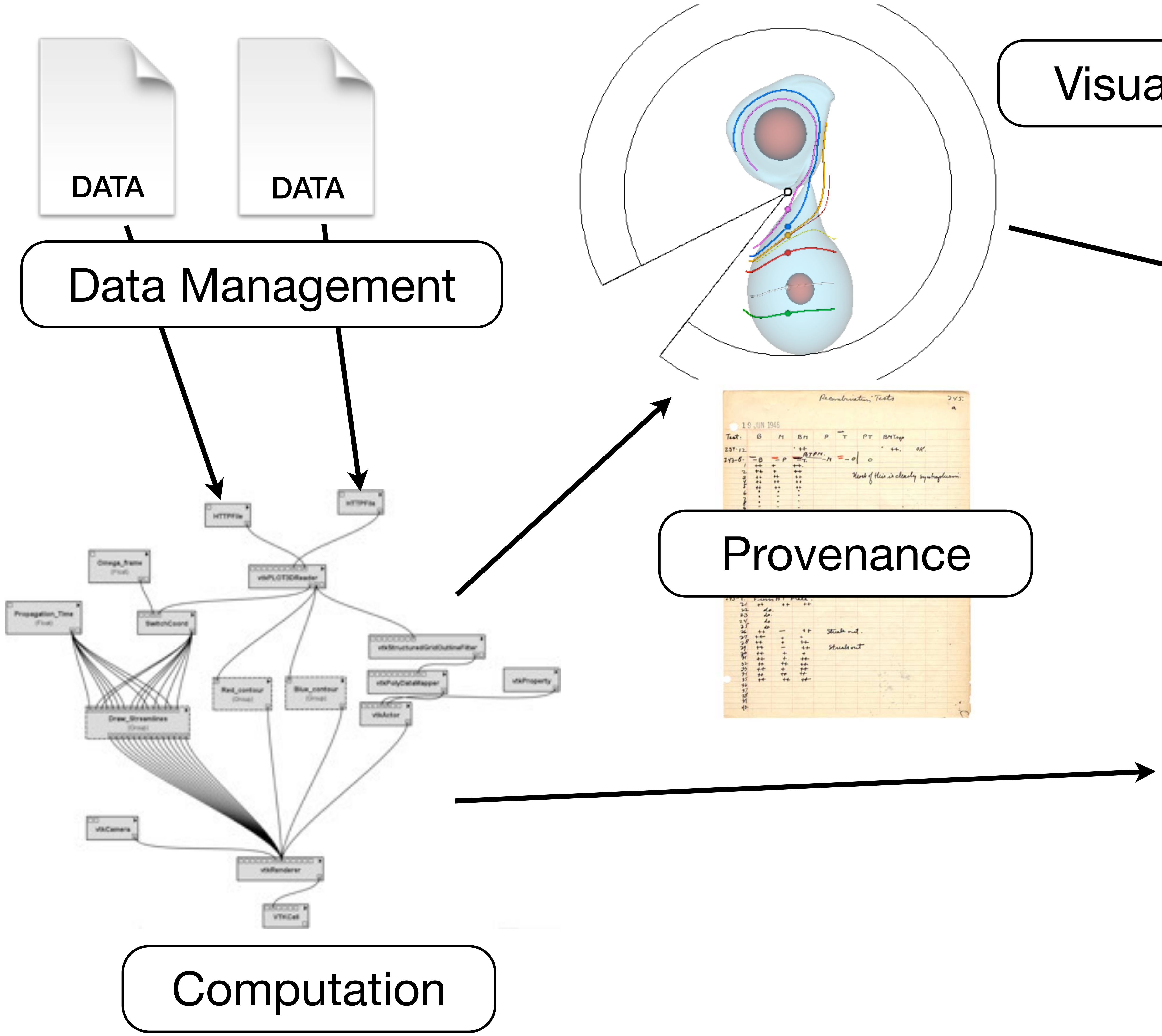
Baton_Rouge,_Louisiana	323
University_of_Mississippi...	230
Mississippi	216
Jackson,_Mississippi	208
Louisiana_State_University...	189
Mississippi_State_University...	169
WVLA-TV	158
Ole_Miss_Rebels_football...	155
List_of_Star_Wars_books...	131
Louisiana	122
New_Orleans_Saints	107



Reno,_Nevada	303
Early_Christianity	284
Comparison_of_the_AK-47_and_M1...	273
Las_Vegas_Academy	225
Timeline_of_Christianity...	216
Las_Vegas	204
Council_of_Jerusalem	192
Paul_the_Apostle	190
University_of_Nevada,_Las_Vega...	189
Nevada	188
Antinomianism	188

[F. Miranda et al., 2017]

Provenance

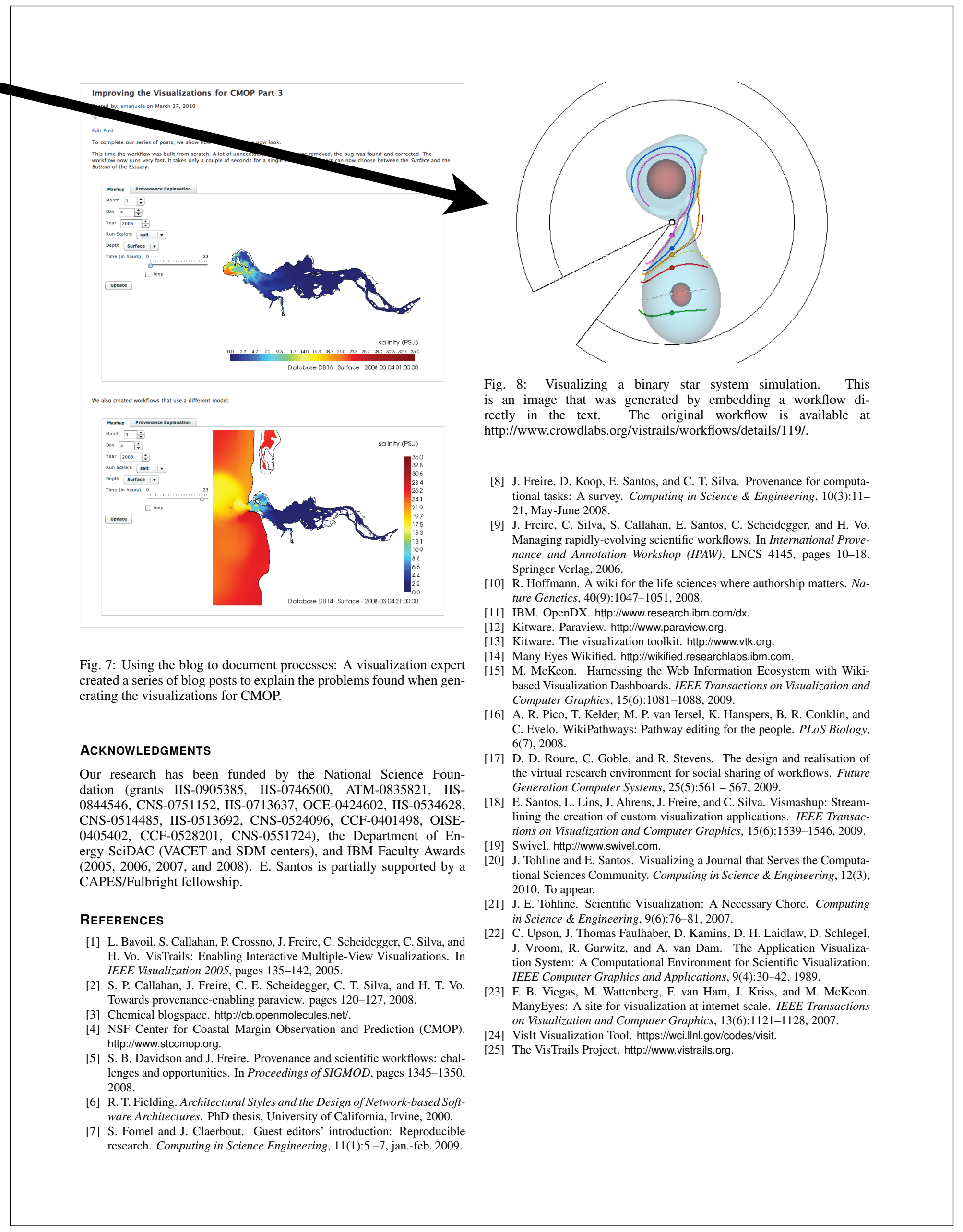


Data Management

Computation

Visualization

Provenance



Improving the Visualizations for CMOP Part 3

Posted by: emanuele on March 27, 2010

Edit Post

To complete our series of posts, we show how the visualizations now look.

This time the workflow was built from scratch. A lot of unnecessary modules were removed, the bug was found and corrected. The workflow now runs very fast. It takes only a couple of seconds for a single timestep. Users can now choose between the *Surface* and the *Bottom* of the Estuary.

We also created workflows that use a different model:

Fig. 7: Using the blog to document processes: A visualization expert created a series of blog posts to explain the problems found when generating the visualizations for CMOP.

ACKNOWLEDGMENTS

Our research has been funded by the National Science Foundation (grants IIS-0905385, IIS-0746500, ATM-0835821, IIS-0844546, CNS-0751152, IIS-0713637, OCE-0424602, IIS-0534628, CNS-0514485, IIS-0513692, CNS-0524096, CCF-0401498, OISE-0405402, CCF-0528201, CNS-0551724), the Department of Energy SciDAC (VACET and SDM centers), and IBM Faculty Awards (2005, 2006, 2007, and 2008). E. Santos is partially supported by a CAPES/Fulbright fellowship.

REFERENCES

[1] L. Bavoil, S. Callahan, P. Crossno, J. Freire, C. Scheidegger, C. Silva, and H. Vo. VisTrails: Enabling Interactive Multiple-View Visualizations. In *IEEE Visualization 2005*, pages 135–142, 2005.

[2] S. P. Callahan, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Towards provenance-enabling paraview. pages 120–127, 2008.

[3] Chemical blogspace. http://cb.openmolecules.net/.

[4] NSF Center for Coastal Margin Observation and Prediction (CMOP). http://www.stccmop.org.

[5] S. B. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *Proceedings of SIGMOD*, pages 1345–1350, 2008.

[6] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[7] S. Fomel and J. Claerbout. Guest editors' introduction: Reproducible research. *Computing in Science Engineering*, 11(1):5 –7, jan.-feb. 2009.

Fig. 8: Visualizing a binary star system simulation. This is an image that was generated by embedding a workflow directly in the text. The original workflow is available at http://www.crowdlabs.org/vistrails/workflows/details/119/.

[8] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21, May-June 2008.

[9] J. Freire, C. Silva, S. Callahan, E. Santos, C. Scheidegger, and H. Vo. Managing rapidly-evolving scientific workflows. In *International Provenance and Annotation Workshop (IPAW)*, LNCS 4145, pages 10–18. Springer Verlag, 2006.

[10] R. Hoffmann. A wiki for the life sciences where authorship matters. *Nature Genetics*, 40(9):1047–1051, 2008.

[11] IBM. OpenDX. http://www.research.ibm.com/dx.

[12] Kitware. Paraview. http://www.paraview.org.

[13] Kitware. The visualization toolkit. http://www.vtk.org.

[14] Many Eyes Wikified. http://wikified.researchlabs.ibm.com.

[15] M. McKeon. Harnessing the Web Information Ecosystem with Wiki-based Visualization Dashboards. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1081–1088, 2009.

[16] A. R. Pico, T. Kelder, M. P. van Iersel, K. Hanspers, B. R. Conklin, and C. Evelo. WikiPathways: Pathway editing for the people. *PLoS Biology*, 6(7), 2008.

[17] D. D. Roure, C. Goble, and R. Stevens. The design and realisation of the virtual research environment for social sharing of workflows. *Future Generation Computer Systems*, 25(5):561 – 567, 2009.

[18] E. Santos, L. Lins, J. Ahrens, J. Freire, and C. Silva. Vismashup: Streamlining the creation of custom visualization applications. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1539–1546, 2009.

[19] Swivel. http://www.swivel.com.

[20] J. Tohline and E. Santos. Visualizing a Journal that Serves the Computational Sciences Community. *Computing in Science & Engineering*, 12(3), 2010. To appear.

[21] J. E. Tohline. Scientific Visualization: A Necessary Chore. *Computing in Science & Engineering*, 9(6):76–81, 2007.

[22] C. Upson, J. Thomas Faulhaber, D. Kamins, D. H. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, and A. van Dam. The Application Visualization System: A Computational Environment for Scientific Visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, 1989.

[23] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. ManyEyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.

[24] VisIt Visualization Tool. https://wci.llnl.gov/codes/visit.

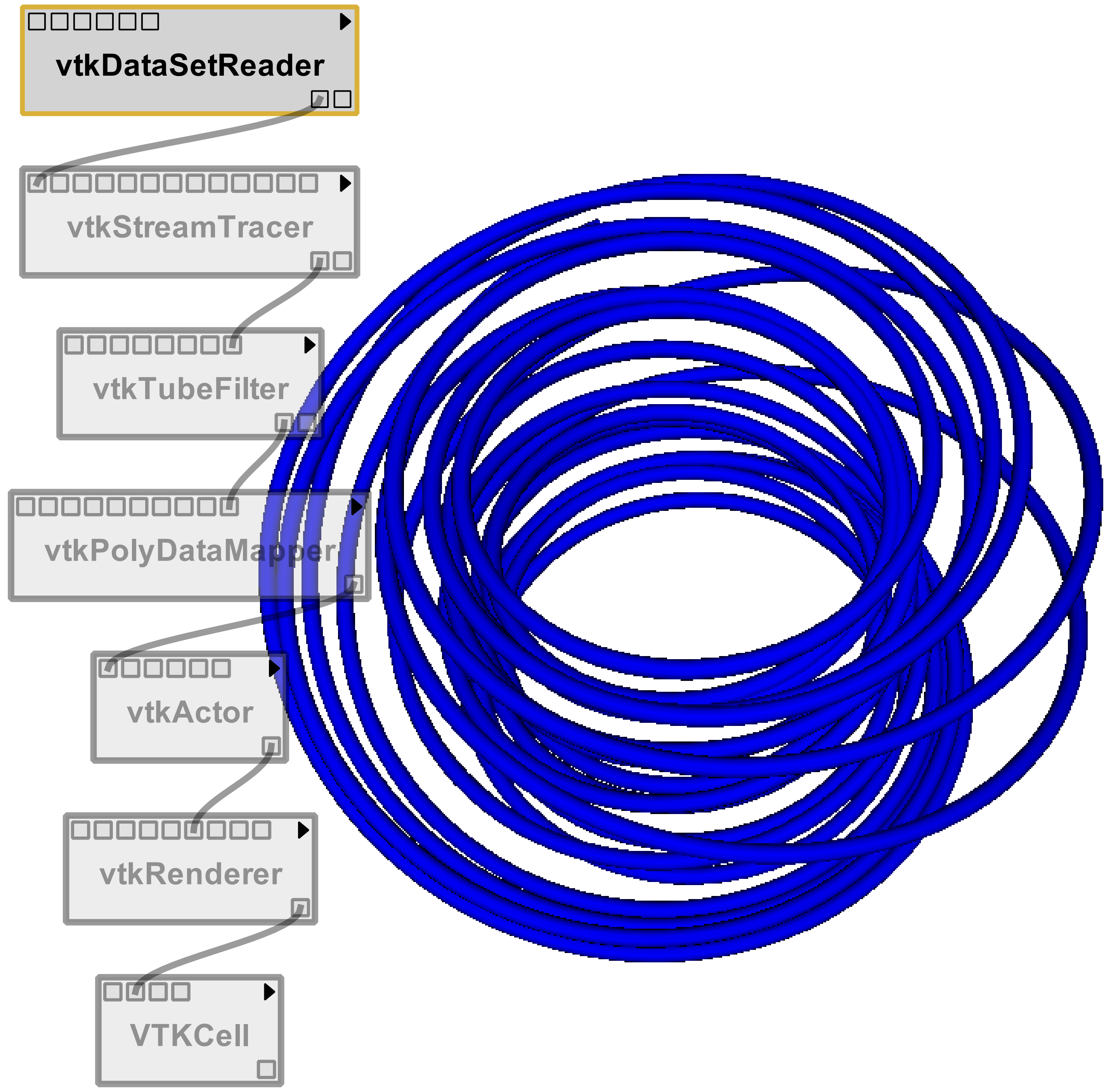
[25] The VisTrails Project. http://www.vistrails.org.

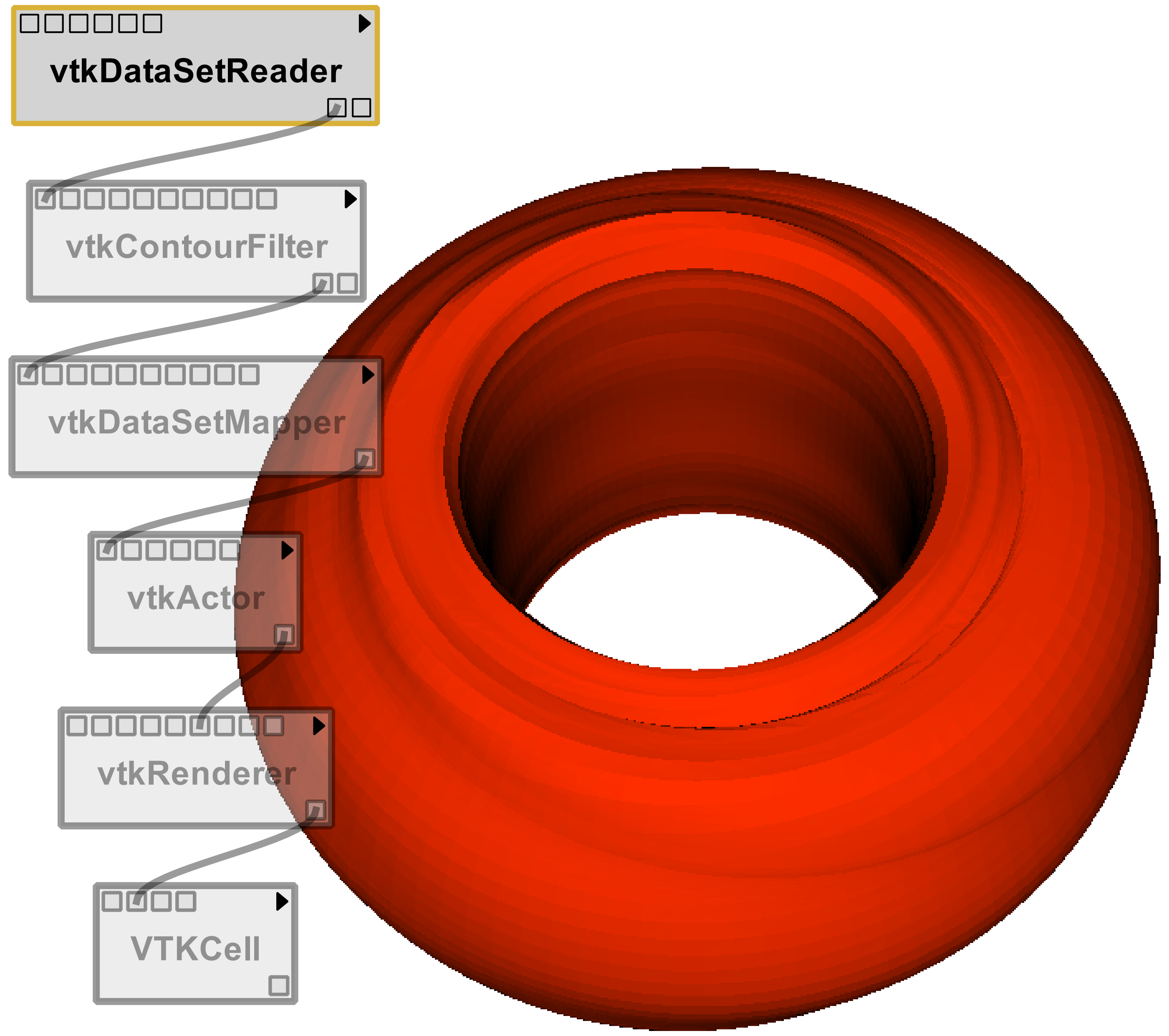
Publishing

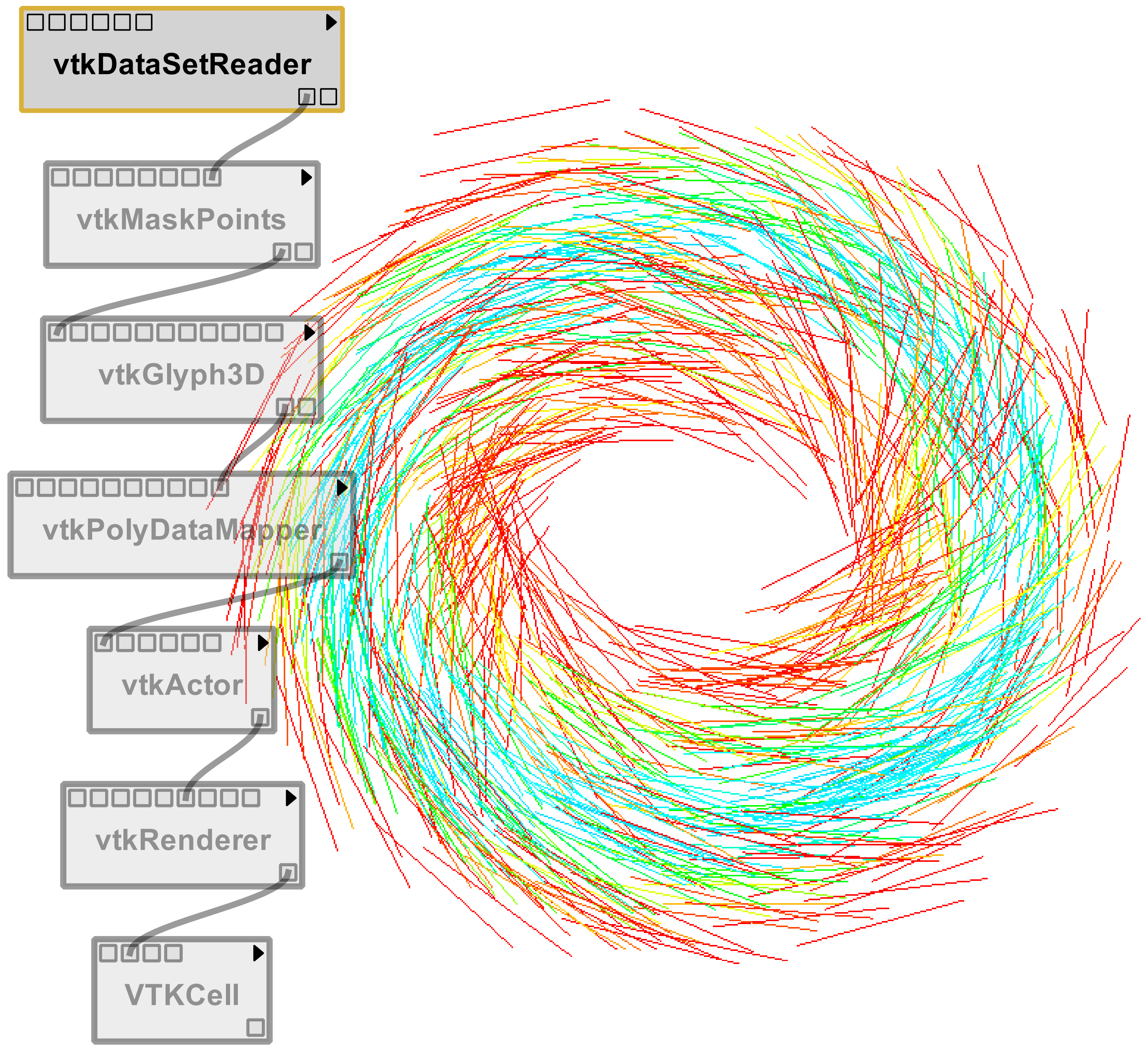
Prospective and Retrospective Provenance

- Recipe for baking a cake versus the actual process & outcome
- Prospective provenance is what was specified/intended
    - a workflow, script, list of steps
- Retrospective provenance is what actually happened
    - actual data, actual parameters, errors that occurred, timestamps, machine information
- **Do not need** prospective provenance to have retrospective provenance!

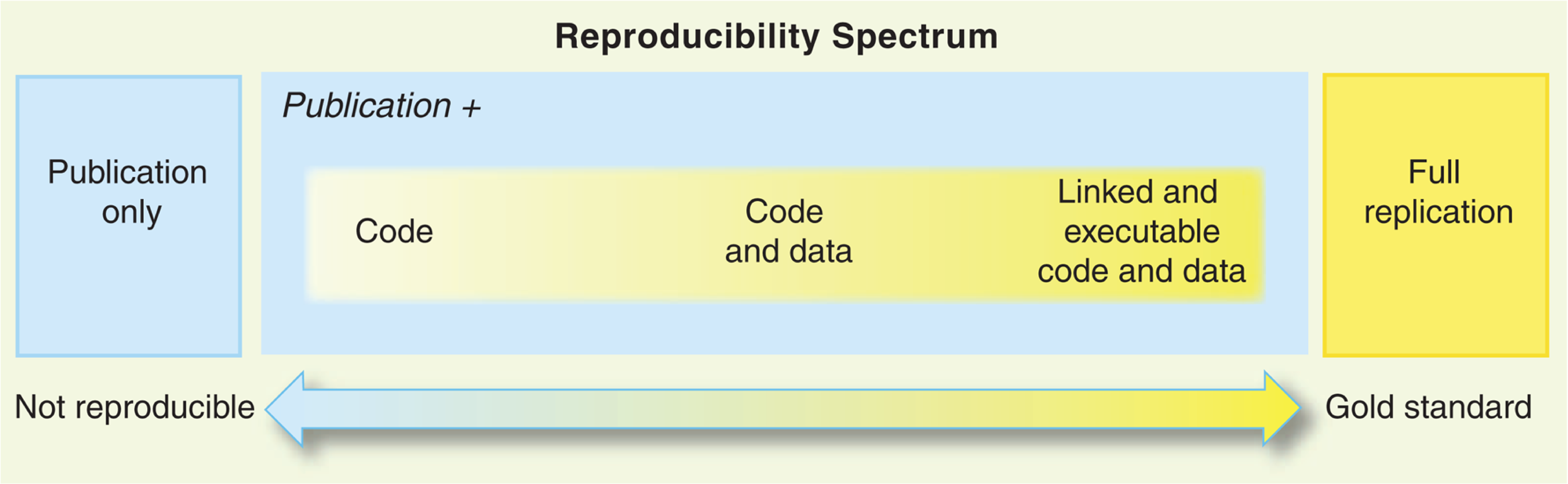
Using Provenance





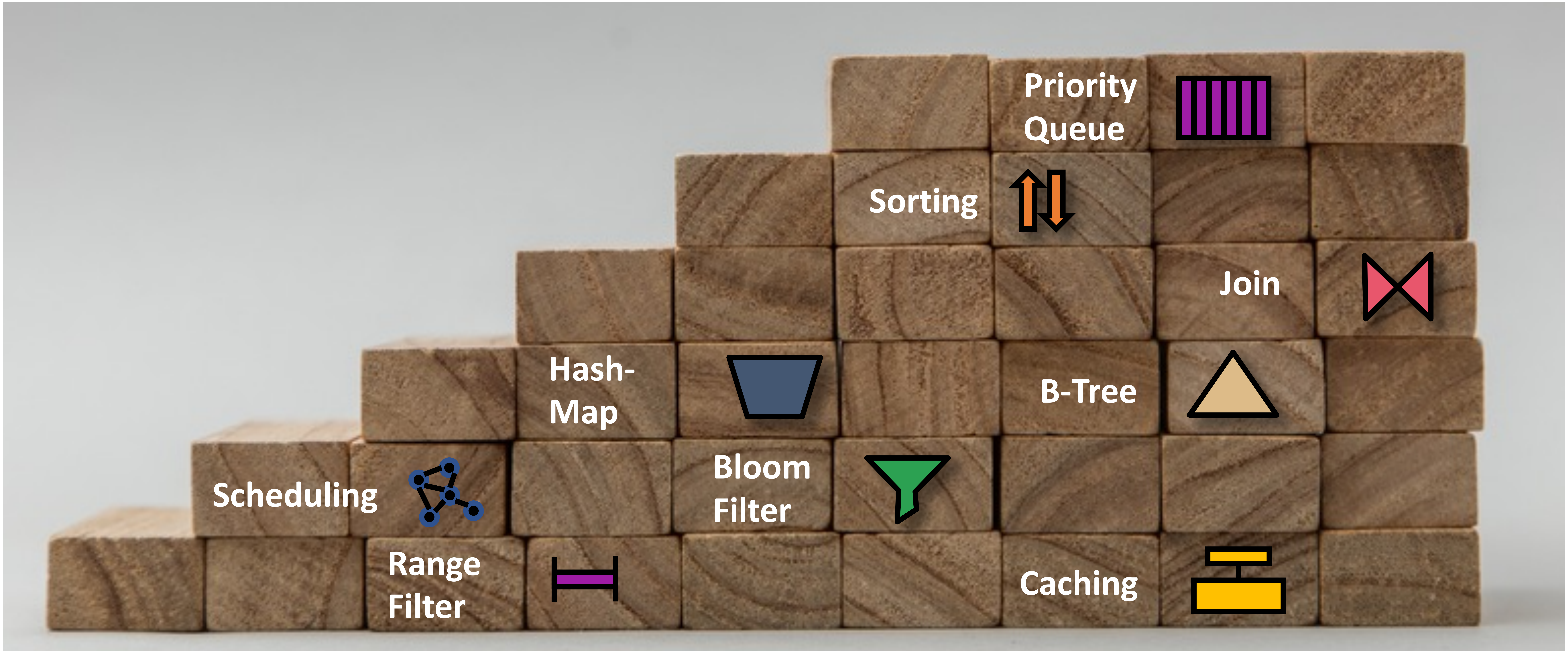


Reproducibility



[R. D. Peng]

Machine Learning and Databases



[T. Kraska, 2019]

Final Exam

- Tuesday, May 5 from 4-5:50pm
- Online
- Similar format to Test 2
- Comprehensive but with more focus on last few weeks of class
- Contact me with questions:
  - Email
  - Setup a time to talk via Blackboard

Stay Safe