Slides Credit: Matthias Boehm

Data Integration and Large Scale Analysis
06 Data Cleaning

Shafaq Siddiqi

Graz University of Technology, Austria

Last update: Nov 10, 2023

Announcement

- **No Lecture on 24th November**
  - Shift every lecture a week forward and last lecture on **26 Jan**
  - **OR**
  - Combine last two lecture and finish on **19 Jan**

Agenda

- **Motivation and Terminology**
- **Data Cleaning and Fusion**
- **Missing Value Imputation**

Motivation and Terminology

Recap: Corrupted/Inconsistent Data

- **#1 Heterogeneity of Data Sources**
  - Update anomalies on denormalized data / eventual consistency
  - Changes of app/prep over time (US vs us) → inconsistencies
- **#2 Human Error**
  - Errors in semi-manual data collection, laziness (see default values), bias
  - Errors in data labeling (especially if large-scale: crowd workers / users)
- **#3 Measurement/Processing Errors**
  - Unreliable HW/SW and measurement equipment (e.g., batteries)
  - Harsh environments (temperature, movement) → aging

Uniqueness & duplicates | **Contradictions & wrong values** | **Missing Values** | **Ref. Integrity** | [**Credit:** Felix Naumann]

<u>ID</u>	Name	BDay	Age	Sex	Phone	Zip
3	Smith, Jane	05/06/1975	44	F	999-9999	98120
3	John Smith	38/12/1963	55	M	867-4511	11111
7	Jane Smith	05/06/1975	24	F	567-3211	98120

Zip	City
98120	San Jose
90001	Lost Angeles

Typos

Examples (aka errors are everywhere)

- **Duplicates**
- **Formatting**
- **Data Entry Errors**
- **Encoding errors**
- **Missing values**
- **Date-time encoding**

```
- US,DFW,LIT,ER4;M83;M83
+ US,DFW,LIT,ER4;M83
```

```
- Beni Airport,Beni,Congo (Kinshasa),BNC,FZNP,0.575,2
+ Beni Airport,Beni,Democratic Republic of Congo,BNC,
```

```
- RAF St Athan,4Q,STN,UNited Kingdom,N
+ RAF St Athan,4Q,STN,United Kingdom,N
```

```
- Oyo Ollombo Airport,Oyo,Congo (Brazzaville),O
+ Oyo Ollombo Airport,Oyo,Republic of Congo,OLL
```

```
ID,NAME,RATING,PHONENUMBER,NO_OF_REVIEWS,ADDRESS
1445980000001,1,5,"(800) 586-5735",38,"867 N Hermitage Ave, Chicago, IL 60622"
1445980000002,326,3.5,"(323) 549-2156",33,"6333 3rd St, Los Angeles, CA 90036"
1445980000003,1760,4,"(415) 359-1212",454,"1760 Polk St, San Francisco, CA 94109"
1445980000004,"□□ ",4,"(773) 866-9898",185,"2977 N Elston Ave, Chicago, IL 60618"
1445980000005,"□□□Disiac Lounge ",3.5,"(212) 586-9880",164,"402 W 54th St, New York, NY 10019"
1445980000006,"□□□Ģ T.'s review of Belly Good Cafe & Crepe",4.5,"(415) 346-8383",843,"1737 Post St, 
1445980000007,"□□ireTrea ",4,"(415) 967-2726",63,"San Francisco, CA 94109"
1445980000008,"10e Restaurant ",4,"(213) 488-1096",166,"811 W 7th St, Los Angeles, CA 90017"
1445980000009,"10th & Wood ",4,"(510) 645-1955",275,"945 Wood St, Oakland, CA 94607"
```

src	flight	scheduled_dept	actual_dept
ua	2011-12-01-UA-2708-EWR-CLT	Thu- Dec 1 2:55 PM	Thu- Dec 1 2:55 PM
airtravelcenter	2011-12-01-UA-2708-EWR-CLT		12/1/11 3:04 PM (-05:00)
myrateplan	2011-12-01-UA-2708-EWR-CLT		12/1/11 3:04 PM (-05:00)
helloflight	2011-12-01-UA-2708-EWR-CLT		12/1/11 3:04 PM (-05:00)
flytecomm	2011-12-01-UA-2708-EWR-CLT		12/1/11 3:04 PM (-05:00)
flights	2011-12-01-UA-2708-EWR-CLT		2011-12-01 02:52 PM
businesstravellogue	2011-12-01-UA-2708-EWR-CLT		2011-12-01 02:52 PM
flylouisville	2011-12-01-UA-2708-EWR-CLT		2011-12-01 02:52 PM
flightstats	2011-12-01-UA-2708-EWR-CLT	2011-12-01 2:55 PM	2011-12-01 2:52 PM
quicktrip	2011-12-01-UA-2708-EWR-CLT	2011-12-01 2:55 PM	2011-12-01 2:52 PM
flightview	2011-12-01-UA-2708-EWR-CLT		3:04 PMDec 01
panynj	2011-12-01-UA-2708-EWR-CLT		3:04 PMDec 01
gofox	2011-12-01-UA-2708-EWR-CLT		3:04 PMDec 01

Terminology

- **#1 Data Cleaning** (aka Data Cleansing)
  - **Detection** and **repair** of data errors
  - **Outliers/anomalies:** values or objects that do not match normal behavior (different goals: data cleaning vs finding interesting patterns)
  - **Data Fusion:** resolution of inconsistencies and errors (e.g., entity resolution **see Lecture 05**)
- **#2 Missing Value Imputation**
  - **Fill missing info** with "best guess"
  - Difference between NAs and 0 (or special values like NaN) for ML models
- **#3 Data Wrangling**
  - Automatic cleaning unrealistic? → Interactive data transformations
  - Recommended transforms + user selection
- **Note:** Partial Overlap w/ KDDM ➔ **it's fine**, different perspectives

Express Expectations as Validity Constraints

- **Manual Approach: "Common Sense"**
- **(Semi-)Automatic Approach: Expectations!**
  - PK → Values must be unique and defined (not null)
  - Exact PK-FK → Inclusion dependencies
  - Noisy PK-FK → Robust inclusion dependencies |R[X]∈S[Y]| / |R[X]| > δ
  - Semantics of attributes → Value ranges / # distinct values
  - Invariant to capitalization
    → Duplicates that differ in capitalization
  - Patterns → regular expressions

Route **Planes**

(Airline, From, To)

```
- US,DFW,LIT,ER4;M83;M83
+ US,DFW,LIT,ER4;M83
```

Age=9999?

```
- RAF St Athan,4Q,STN,UNited Kingdom,N
+ RAF St Athan,4Q,STN,United Kingdom,N
```

2019-11-15 vs Nov 15, 2019

- **Formal Constraints**
  - Functional dependencies (FD), conditional FDs (CFD), metric dependencies
  - Inclusion dependencies, matching dependencies
  - Denial constraints

$$\forall t_\alpha t_\beta \in R: \neg(t_\alpha.Role = t_\beta.Role \wedge t_\alpha.City = 'NYC' \wedge t_\beta.City \neq 'NYC' \wedge t_\alpha.Salary < t_\beta.Salary)$$

Data Cleaning and Fusion

Data Validation

validation checks on expected shape
before training first model

[Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, Martin Zinkevich: Data Management Challenges in Production Machine Learning. Tutorial, **SIGMOD 2017**]


Google
Data Management Challenges in Production Machine Learning
Neoklis Polyzotis, Sudip Roy, Steven Whang, Martin Zinkevich

(**Google Research**)

- **Check a feature's min, max, and most common value**
  - Ex: Latitude values must be within the range [-90, 90] or [-π/2, π/2]
- **The histograms of continuous or categorical values are as expected**
  - Ex: There are similar numbers of positive and negative labels
- **Whether a feature is present in enough examples**
  - Ex: Country code must be in at least 70% of the examples
- **Whether a feature has the right number of values (i.e., cardinality)**
  - Ex: There cannot be more than one age of a person

Data Validation, cont.

[Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Bießmann, Andreas Grafberger: Automating Large-Scale Data Quality Verification. **PVLDB 2018**]

(**Amazon Research**)

- **Constraints and Metrics for quality check UDFs**

constraint	arguments
dimension *completeness*	
isComplete	column
hasCompleteness	column, udf
dimension *consistency*	
isUnique	column
hasUniqueness	column, udf
hasDistinctness	column, udf
isInRange	column, value range
hasConsistentType	column
isNonNegative	column
isLessThan	column pair
satisfies	predicate
satisfiesIf	predicate pair
hasPredictability	column, column(s), udf
statistics (can be used to verify dimension *consistency*)	
hasSize	udf
hasTypeConsistency	column, udf
hasCountDistinct	column
hasApproxCountDistinct	column, udf
hasMin	column, udf
hasMax	column, udf
hasMean	column, udf
hasStandardDeviation	column, udf
hasApproxQuantile	column, quantile, udf
hasEntropy	column, udf
hasMutualInformation	column pair, udf
hasHistogramValues	column, udf
hasCorrelation	column pair, udf
time	
hasNoAnomalies	metric, detector

metric
dimension *completeness*
Completeness
dimension *consistency*
Size
Compliance
Uniqueness
Distinctness
ValueRange
DataType
Predictability
statistics (can be used to
Minimum
Maximum
Mean
StandardDeviation
CountDistinct
ApproxCountDistinct
ApproxQuantile
Correlation
Entropy
Histogram
MutualInformation

Organizational Lesson: benefit of shared vocabulary/procedures

Technical Lesson: fast/scalable; reduce manual and ad-hoc analysis

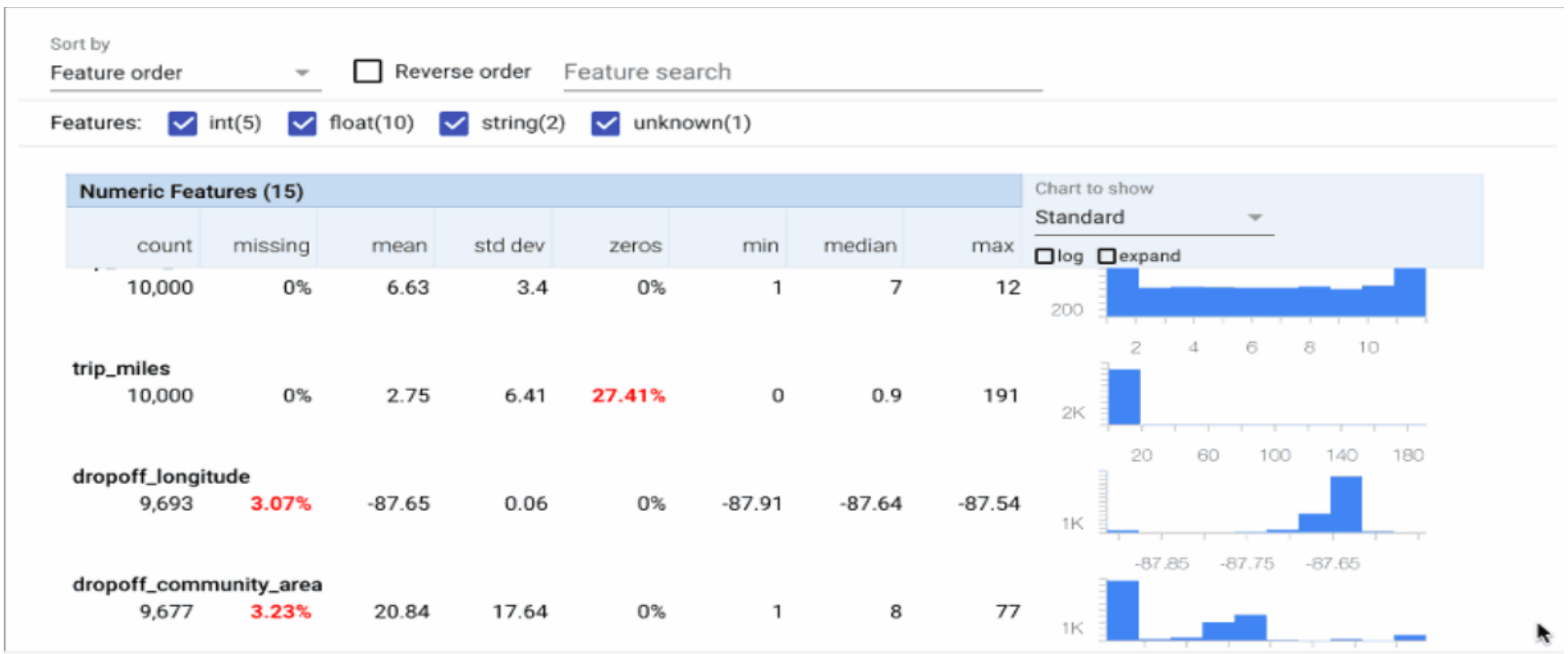
- **Approach**
  - **#1** Quality checks on basic metrics, computed in **Apache Spark**
  - **#2** **Incremental maintenance** of metrics and quality checks

Data Validation, cont.

[Mike Dreves; Gene Huang; Zhuo Peng; Neoklis Polyzotis; Evan Rosen; Paul Suganthan: From Data to Models and Back. **DEEM 2020**]

(**Google**)

- **TensorFlow Data Validation (TFDV)**
  - Library or TFX components
  - Provides functions for stats computation, validation checks and anomaly detection



Standardization and Normalization

- **#1 Standardization**
  - Centering and scaling to mean 0 and variance 1
  - **Ensures well-behaved training**
  - **Densifying operation**
  - Awareness of NaNs
  - Batch normalization in DNN: standardization of activations

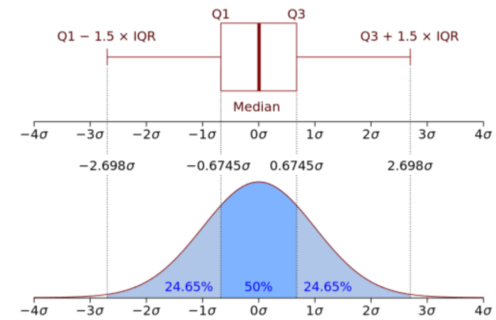
```
X = X - colMeans(X);
X = X / sqrt(colVars(X));

X = replace(X, pattern=NaN,
  replacement=0); #robustness
```

- **#2 Normalization**
  - Aka min-max normalization
  - Rescale values into common range [0,1]
  - **Avoid bias to large-scale features**
  - Does not handle outliers

```
X = (X - colMins(X))
  / (colMaxs(X) - colMins(X));
```

Winsorizing and Trimming



[**Credit:** https://en.wikipedia.org]

- **Recap: Quantiles**
  - Quantile Q_p w/ $p \in (0,1)$ defined as $P[X \leq x] = p$

- **Winsorizing**
  - **Replace** tails of data distribution at user-specified threshold
  - Quantiles / std-dev
  - ➔ Reduce skew

```
# compute quantiles for lower and upper
ql = quantile(X, 0.05);
qu = quantile(X, 0.95);

# replace values outside [ql,qu] w/ ql and qu
Y = ifelse(X < ql, ql, X);
Y = ifelse(Y > qu, qu, Y);
```

SystemDS:
`winsorize()`
`outlier()`

- **Truncation/Trimming**
  - **Remove** tails of data distribution at user-specified threshold

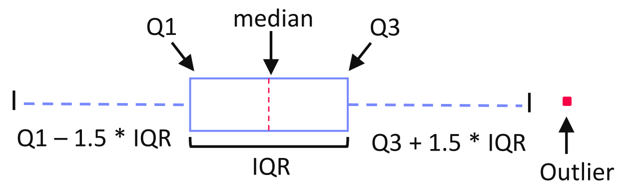
```
# remove values outside [ql,qu]
I = X < qu | X > ql;
Y = removeEmpty(X, "rows", select = I);
```

- **Largest Difference from Mean**

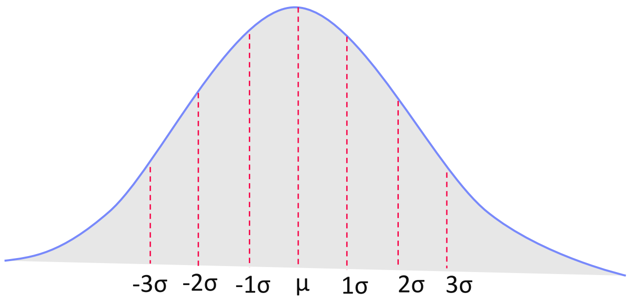
```
# determine largest diff from mean
I = (colMaxs(X)-colMeans(X))
  > (colMeans(X)-colMins(X));
Y = ifelse(xor(I,op), colMaxs(X), colMins(X));
```

Winsorizing and Trimming, cont.

- **SystemDS outlierByIQR**
  - less than Q1 – (k × IQR) or greater than Q3 + (k × IQR) → **outlier**



- **SystemDS outlierBySd**
  - less than mean – (k × stdev) or greater than mean + (k × stdev) → **outlier**



- **Methods for Handling Outliers**
  - Replace outliers with default values (constants or mean/median/mode)
  - Update outliers as missing values
  - Data clipping

Outliers and Outlier Detection

- **Types of Outliers**
  - **Point outliers:** single data points far from the data distribution
  - **Contextual outliers:** noise or other systematic anomalies in data
  - **Sequence (contextual) outliers:** sequence of values w/ abnormal shape/agg
  - Univariate vs multivariate analysis
  - Beware of underlying assumptions (distributions)

[Varun Chandola, Arindam Banerjee, Vipin Kumar: Anomaly detection: A survey. **ACM Comput. Surv. 2009**]

- **Types of Outlier Detection**
  - **Type 1 Unsupervised:** No prior knowledge of data, similar to unsupervised **clustering** → **expectations:** distance, # errors
  - **Type 2 Supervised:** Labeled normal and abnormal data, similar to supervised **classification**
  - **Type 3 Normal Model:** Represent normal behavior, similar to **pattern recognition** → **expectations:** rules/constraints

[Victoria J. Hodge, Jim Austin: A Survey of Outlier Detection Methodologies. **Artif. Intell. Rev. 2004**]

Outlier Detection Techniques

[Varun Chandola, Arindam Banerjee, Vipin Kumar: Anomaly detection: A survey. **ACM Comput. Surv. 2009**]

- **Classification**
  - Learn a classifier using labeled data
  - Binary: normal / abnormal
  - Multi-class: k normal / abnormal (one against the rest) → none=abnormal
  - **Examples: AutoEncoders, Bayesian Networks, SVM, decision trees**
- **K-Nearest Neighbors**
  - Anomaly score: distance to kth nearest neighbor
  - Compare distance to threshold + (optional) max number of outliers
- **Clustering**
  - Clustering of data points, anomalies are points not assigned / too far away
  - **Examples: DBSCAN** (density), **K-means** (partitioning)
  - Cluster-based local outlier factor (global, local, and size-specific density)

Outlier Detection Techniques, cont.

- **Frequent Itemset Mining**
  - Rare itemset mining / sequence mining; Examples: Apriori/Eclat/FP-Growth

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Coverage Analysis**
  - Given a database D and a data pattern P
  - Coverage of a data pattern cov(P) is defined as the number of records in table T that satisfy pattern P
  - Pattern P is a covered pattern if cov(P) ≥ τ
  - Otherwise, this pattern is said to be uncovered

[Yin Lin et al: Identifying Insufficient Data Coverage in Databases with multiple Relations. **PVLDB 2020**]

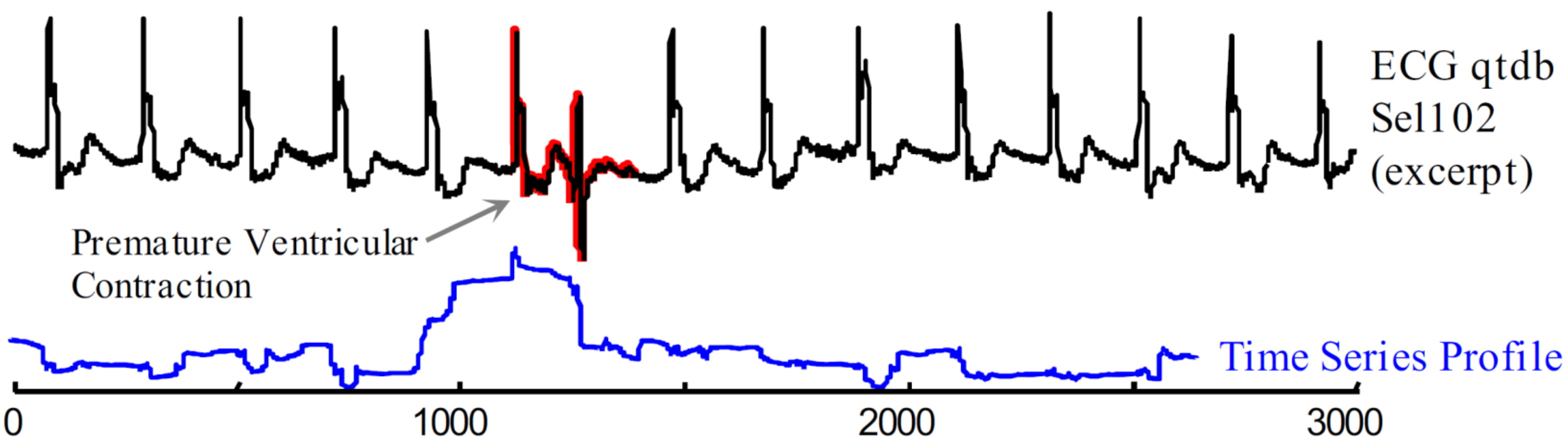
Time Series Anomaly Detection

- **Basic Problem Formulation**
  - Given regular (equi-distant) time series of measurements
  - Detect anomalous subsequences s of **length l** (fixed/variable)

- **Anomaly Detection**
  - #1 Supervised: **Classification problem**
  - #2 Unsupervised: **k-Nearest Neighbors** (discords) → All-pairs similarity join

[**Matrix Profile XXVIII**, **SDM 2023**]

[Chin-Chia Michael Yeh et al: **Matrix Profile I**: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. **ICDM 2016**]



Outlier Detection in Non-IID Data

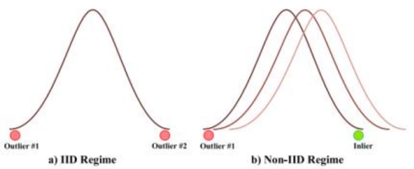


FIGURE 1. Changing definition of outliers in concept drift.

- **Non-Independent and Identically Distributed (non-IID)**
  - Inter-dependencies, correlations, heterogeneity, and non-stationarity
  - Indicating coupling, correlations between variables

- **ARCUS (Adaptive framework foR online deep anomaly deteCtion Under a complex evolving data Stream)**
  - A model pool of auto-encoders
  - Same structure but different hyperparameters
  - Concept drift aware pool adaption using Hoeffding's Inequality (statistical test)

[Susik Yoon et. al. Adaptive Model Pooling for Online Deep Anomaly Detection from a Complex Evolving Data Stream. **KDD 2022**]

https://datasciences.org/non-iid-learning/

Automatic Data Repairs

- **Overview Repairs**
  - Question: Repair data, rules/constraints, or both?
  - General principle: **"minimality of repairs"**

- **Example Data Repair**
  - Functional dependency **A→ B**
  - Violation for A=1

[Xu Chu, Ihab F. Ilyas: Qualitative Data Cleaning. Tutorial, **PVLDB 2016**]

A	B
1	2
1	3
1	3
4	5

OK, dist=1

A	B
1	**3**
1	3
1	3
4	5

vs

A	B
1	2
1	**2**
1	**2**
4	5

vs

A	B
1	**5**
1	**5**
1	**5**
4	5

- **Note: Piece-meal vs holistic data repairs**

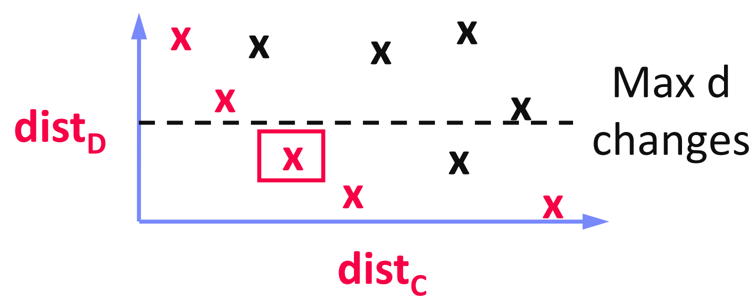
Automatic Data/Rule Repairs, cont.

- **Example**
  - Expectation: **City → Country**; new data conflicts

[George Beskales, Ihab F. Ilyas, Lukasz Golab, Artur Galiullin: On the relative trust between inconsistent data and inaccurate constraints. **ICDE 2013**]

IATA	ICAO	Name	City	Country
MEL	YMML	Melbourne International Airport	Melbourne	Australia
MLB	KMLB	Melbourne International Airport	Melbourne	USA

- **Relative Trust: {FName, LName} → Salary**
  - **Trusted FD:** ➔ change salary according to {FName, LName} → Salary
  - **Trusted Data:** ➔ change FD to {FName, LName, DoB, Phone} → Salary
  - **Equally-trusted:** ➔ change FD to {FName, LName, DoB} → Salary AND data accordingly



Excursus: Simpson's Paradox

- **Overview:** Statistical paradox stating that an analysis of groups may yield **different results at different aggregation levels**

- **Example UC Berkeley '73**

	Applicants	Admitted
Men	8442	**44%**
Women	4321	35%

➔ more women had applied to departments that admitted a small percentage of applicants

	Men		Women	
	Appl.	Adm.	Appl.	Adm.
A	**825**	62%	108	82%
B	**560**	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	373	6%	341	7%

"**The real Berkeley story**
A Wall Street Journal interview with Peter Bickel, one of the statisticians involved in the original study, makes clear that Berkeley was never sued—it was merely afraid of being sued"

[https://www.refsmmat.com/posts/2016-05-08-simpsons-paradox-berkeley.html]

Selected Research

[Jiannan Wang et al: A sample-and-clean framework for fast and accurate query processing on dirty data. **SIGMOD 2014**]

[Sanjay Krishnan et al: ActiveClean: Interactive Data Cleaning For Statistical Modeling. **PVLDB 2016**]

- **ActiveClean (SampleClean)**
  - Suggest sample of data for manual cleaning (rule/ML-based detectors, **Simpson's paradox**)
  - **Example Linear Regression**

  - **Approach:** Cleaning and training as form of SGD
    - Initialization: model on dirty data
    - Suggest sample of data for cleaning
    - Compute gradients over newly cleaned data
    - Incrementally update model w/ weighted gradients of previous steps

Selected Research, cont.

- **HoloClean**
  - Clean and enrich based on quality rules, value correlations, and reference data
  - Probabilistic models for capturing data generation
  - **HoloDetect**
    - **Learn data representations** of errors
    - **Data augmentation** w/ erroneous data from sample of clean data (add/remove/exchange characters)

[Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, Christopher Ré: HoloClean: Holistic Data Repairs with Probabilistic Inference. **PVLDB 2017**]

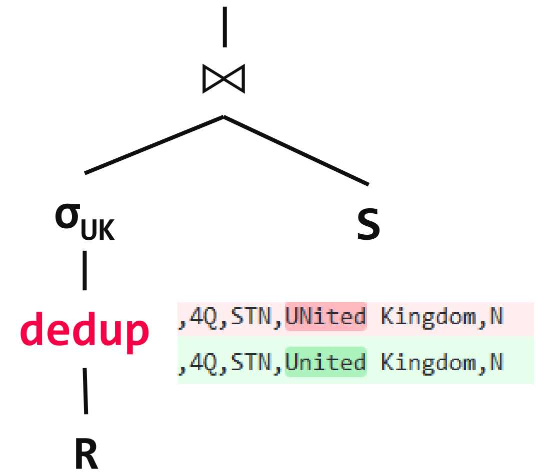
[Alireza Heidari, Joshua McGrath, Ihab F. Ilyas, Theodoros Rekatsinas: HoloDetect: Few-Shot Learning for Error Detection, **SIGMOD 2019**]

- **Other Systems**
  - **AlphaClean** (generate data cleaning pipelines) [preprint 2019]
  - **BoostClean** (generate repairs for domain value violations) [preprint 2017]
  - **CPClean** (prioritize repairs on incomplete data)[Bojan Karlaš et al. PVLDB 2021]

Query Planning w/ Data Cleaning

[Dong Deng et al: The Data Civilizer System. **CIDR 2017**]

- **Problem**
  - Given query tree or data flow graph
  - Find placement of data cleaning operators to reduce costs
- **Approach**
  - Budget B of user actions
  - Active learning user feedback on query results
  - Map query results back to sources via lineage
  - Cleaning in decreasing order of impact



- **Extensions?**
  - **Query-aware placement/refinement** (e.g., UK) of cleaning primitives
  - **Ordering of cleaning primitives** (norm, dedup, missing value?)

Data Wrangling

- **Data Wrangler Overview**
  - Interactive data cleaning via spreadsheet-like interfaces
  - Iterative structure inference, recommendations, and data transformations
  - Predictive interaction (infer next steps from interaction)

[Vijayshankar Raman, Joseph M. Hellerstein: Potter's Wheel: An Interactive Data Cleaning System. **VLDB 2001**]

[Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, Jeffrey Heer: Wrangler: interactive visual specification of data transformation scripts. **CHI 2011**]

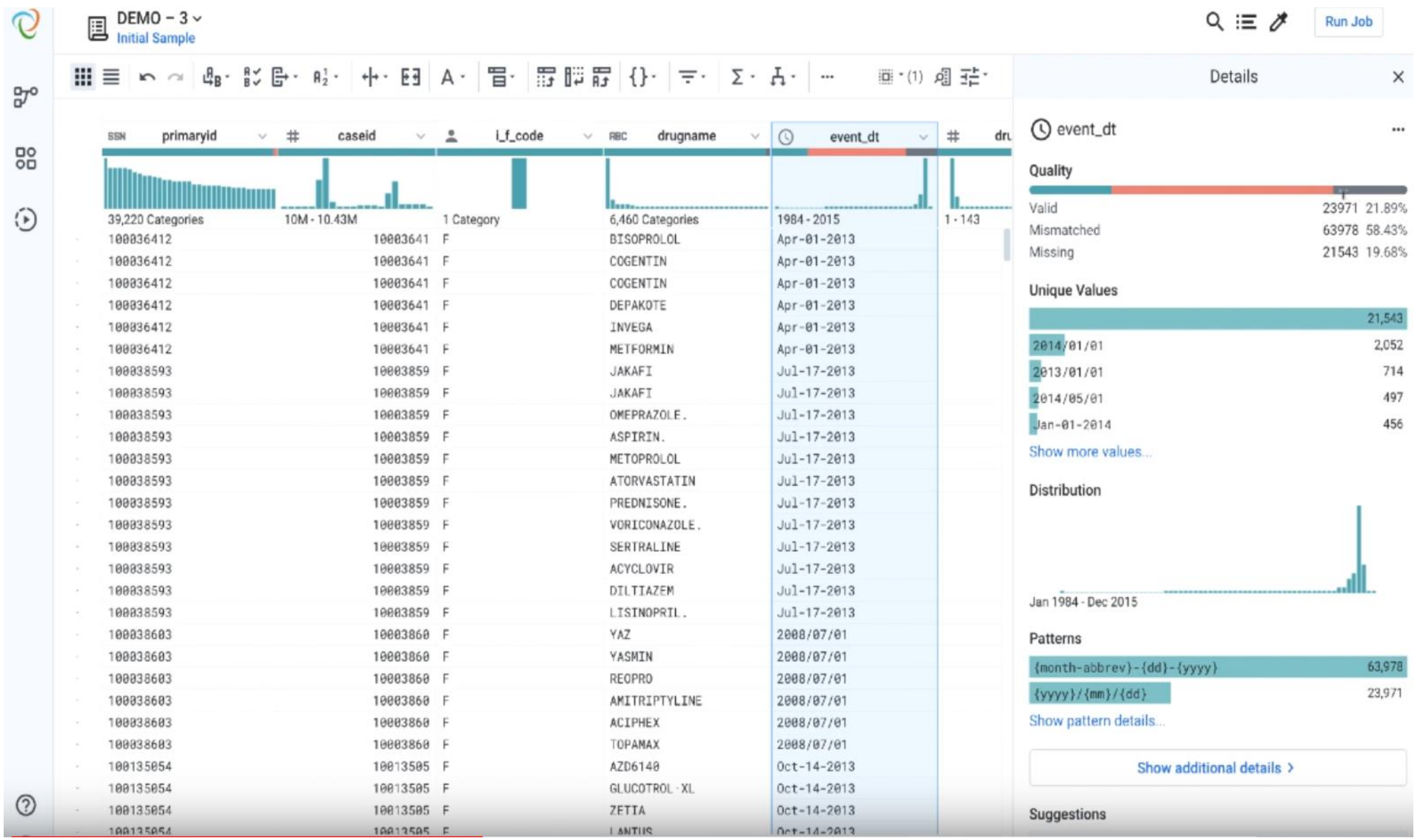
[Jeffrey Heer, Joseph M. Hellerstein, Sean Kandel: Predictive Interaction for Data Transformation. **CIDR 2015**]

- **Commercial/Free Tools**
  - Trifacta (from Data Wrangler)
  - Google Fusion Tables: semi-automatic resolution and deduplication (sunset Dec 2019)

Data Wrangling, cont.

[**Credit:** Alex Chan (Apr 2, 2019) https://www.trifacta.com/blog/trifacta-for-data-quality-introducing-smart-cleaning/]

- **Example: Trifacta Smart Cleaning**



Missing Value Imputation

Basic Missing Value Imputation

- **Missing Value**
  - Application context defines if 0 is missing value or not
  - If differences between 0 and missing values, use NA or NaN?
  - Could be a number outside the domain or symbol as '?'

- **Relationship to Data Cleaning**
  - Missing value is error, need to generate **data repair**
  - Data imputation techniques can be used as **outlier/anomaly detectors**

- **Recap: Reasons**
  - **#1 Heterogeneity of Data Sources**
  - **#2 Human Error**
  - **#3 Measurement/Processing Errors**

→

MCAR: Missing Completely at Random
MAR: Missing at Random
MNAR: Missing Not at Random

Basic Missing Value Imputation

- **Missing Completely at Random**
  - Missing values are randomly distributed across all records (independent from recorded or missing values)

ID	Position	Salary ($)	
1	Manager	**null**	(3500)
2	Secretary	2200	
3	Manager	3600	
4	Technician	**null**	(2400)
5	Technician	2500	
6	Secretary	**null**	(2000)

- **Missing at Random**
  - Missing values are randomly distributed within one or more sub-groups of records
  - Missing values depend on the recorded but not on the missing values, and **can be recovered**

ID	Position	Salary ($)
1	Manager	3500
2	Secretary	2200
3	Manager	3600
4	**Technician**	**null**
5	**Technician**	**null**
6	Secretary	2000

- **Not Missing at Random**
  - Missing data depends on the missing values themselves
  - E.g., missing low salary, age, weight, etc.

ID	Position	Salary ($)
1	Manager	3500
2	Secretary	**null**
3	Manager	3600
4	Technician	**null**
5	Technician	2500
6	Secretary	**null**

<= 2400 missing

[Abdulhakim Ali Qahtan, Ahmed K. Elmagarmid, Raul Castro Fernandez, Mourad Ouzzani, Nan Tang: FAHES: A Robust **Disguised Missing Values** Detector. **KDD 2018**]

Basic Missing Value Imputation, cont.

- **Basic Value Imputation** (for MCAR)
  - **General-purpose:** **replace** by user-specified constant, or **drop records,** or **one-hot encode** as separate column
  - **Continuous variables:** replace by **mean, median**
  - **Categorical variables:** replace by **mode** (most frequent category)

- **Iterative Algorithms** (**chained-equation imputation** for MAR)
  - Train ML model on available data to predict missing information
    - Initialize with basic imputation (e.g., mean)
    - One dirty variable at a time
    - Feature k → label, split data into training: observed / scoring: missing
    - Types: categorical → classification, continuous → regression
  - Noise reduction: train models over feature subsets + averaging

[Stef van Buuren, Karin Groothuis-Oudshoorn: mice: Multivariate Imputation by Chained Equations in R, **J. of Stat. Software 2011**]

Basic Missing Value Imputation, cont.

- **MICE example**
  - Initialization: fill in the missing values with column mean (w/ or w/o NAs)
  - Iterations: each column per iteration

V1	V2	V3	V4	V5
1	56	2	2	2
2	23	0	0	0
1	NA	0	0	2
2	24	-1	2	NA
NA	22	1	2	0

V1	V2	V3	V4	V5
1	56	2	2	2
2	23	0	0	0
1	25	0	0	2
2	24	-1	2	0.8
1.2	22	1	2	0

V1	V2	V3	V4	V5
1	56	2	2	2
2	23	0	0	0
1	25	0	0	2
2	24	-1	2	0.8
1.2	22	1	2	0

train(y) train(x)

V1	V2	V3	V4	V5
1	56	2	2	2
2	23	0	0	0
1	25	0	0	2
2	24	-1	2	0.8
?	22	1	2	0

← test(x)

BREAK (and Test Yourself)

[Exam Feb 08, 2021]

A	B	C	D	E
Red	2100	X	DE	35
Orange	4300	**NULL**	DE	**NULL**
Yellow	5700	Z	DE	35
Green	2500	X	AT	25
Blue	4900	Y	US	**NULL**
Violet	5200	**NULL**	US	45

- **Two techniques for MVI in the categorical column C. If possible, provide the imputed values** (6 points)
  - Mode → {X, X}
  - Functional Dependency (e.g., B/1000→C) → {Y, Z}
  - ML (Classification)
- **Two techniques for MVI in the numerical column E. If possible, provide the imputed values** (6 points)
  - Mean → {35, 35}
  - Functional Dependency (e.g., D→E) → {35, 45}
  - ML (Regression)

DNN Based MV Imputation

[Felix Bießmann et al: DataWig: Missing Value Imputation for Tables, **J. of ML Research 2019**]

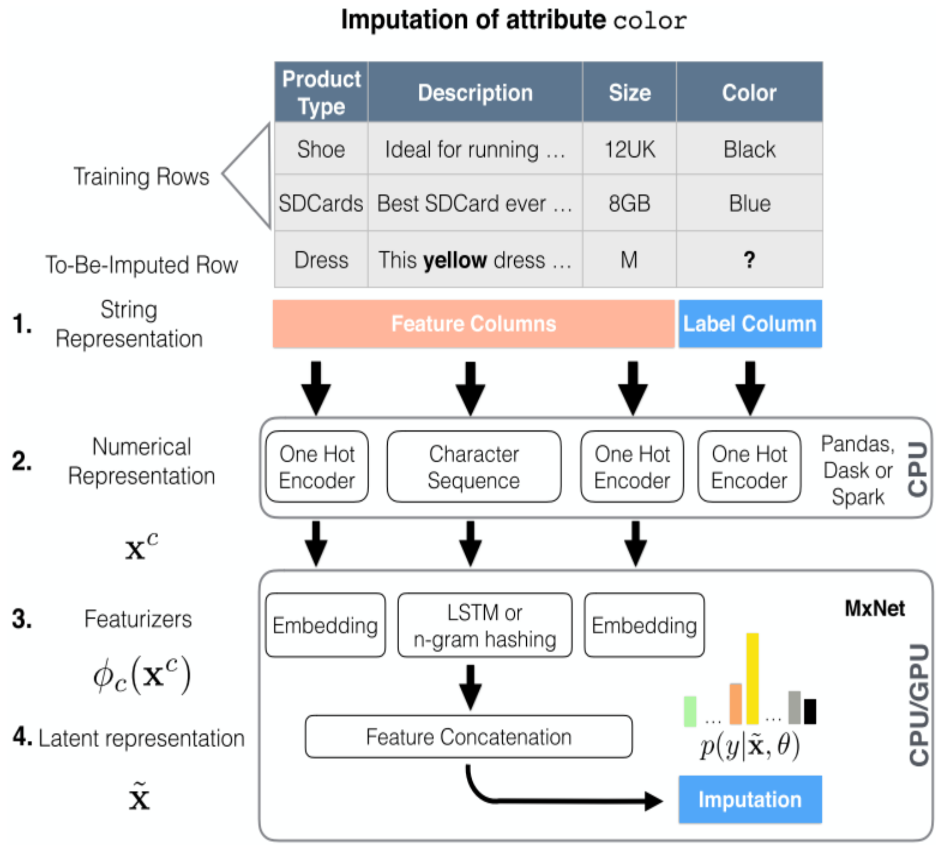
- **DataWig**
  - Missing values imputation for heterogeneous data including unstructured text

Data Type	Featurizers	Loss
Numerical	Normalization Neural Network	Regression
Categorical	Embeddings	Softmax
Text	Bag-of-Words LSTM	N/A

```
table = pandas.read_csv('products.csv')
missing = table[table['color'].isnull()]

# instantiate model and train imputer
model = SimpleImputer(
          input_columns=['description',
                         'product_type',
                         'size'],
          output_columns=['color'])
        .fit(table)

# impute missing values
imputed = model.predict(missing)
```

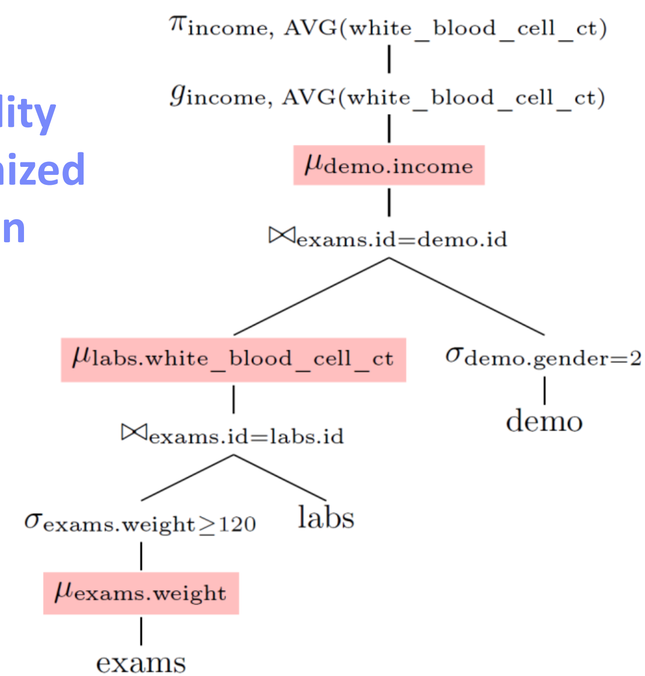


Query Planning w/ MV Imputation

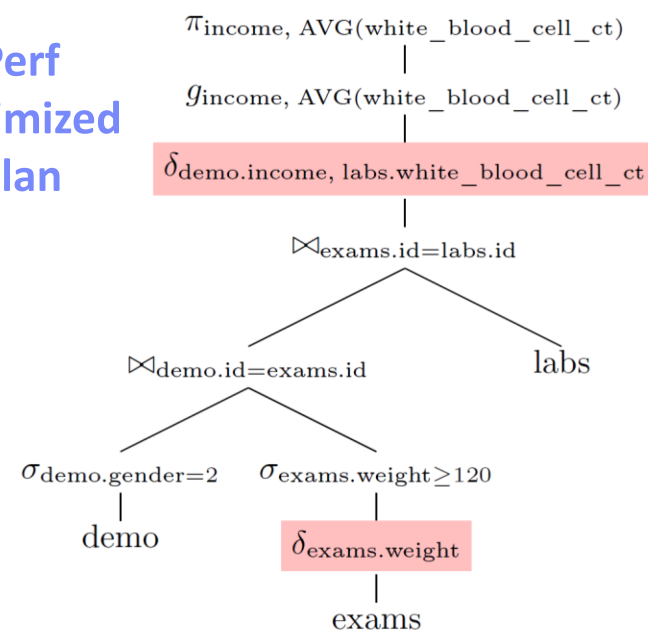
- **Dynamic Imputation**
  - Data exploration w/ on-the-fly imputation
  - Optimal placement of **drop δ** and **impute μ** (**chained-equation imputation** via decision trees)
  - **Multi-objective optimization**

[Jose Cambronero, John K. Feser, Micah Smith, Samuel Madden: Query Optimization for Dynamic Imputation. **PVLDB 2017**]

Quality Optimized Plan



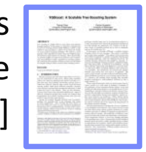
Perf Optimized Plan



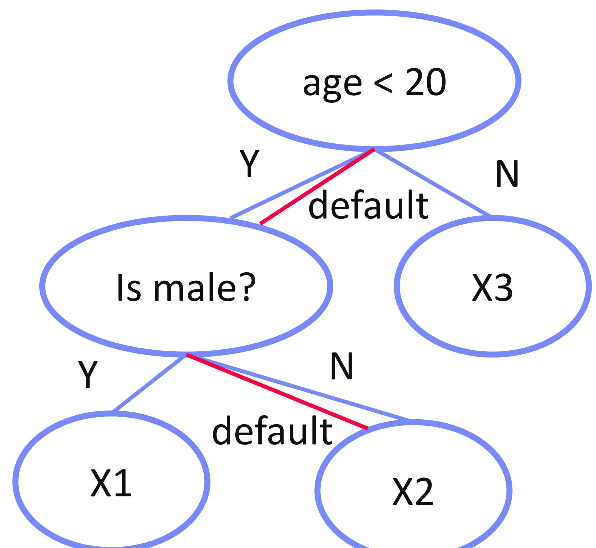
XGBoost's Sparsity-aware Split Finding

- **Motivation**
  - Missing values
  - Sparsity in general
    (zero values, one-hot encoding)
- **XGBoost**
  - Implementation of gradient boosted decision trees
  - Multi-threaded, cache-conscious
- **Sparsity-aware Split Finding**
  - Handles the missing values by **default paths** (learned from data)
  - An example will be classified into the default direction when the feature needed for the split is missing

[Tianqi Chen and Charlos Guestrin: XGBoost: A Scalable Tree Boosting System, **KDD 2016**]



Example	Age	Gender
X1	?	male
X2	15	?
X3	25	female



Time Series Imputation

[Steffen Moritz and Thomas Bartz-Beielstein: imputeTS: Time Series Missing Value Imputation in R, **The R Journal 2017**]

- **Example R Package imputeTS**

Function	Option	Description
na.interpolation		
	linear	Imputation by Linear Interpolation
	spline	Imputation by Spline Interpolation
	stine	Imputation by Stineman Interpolation
na.kalman		
	StructTS	Imputation by Structural Model & Kalman Smoothing
	auto.arima	Imputation by ARIMA State Space Representation & Kalman Sm.
na.locf		
	locf	Imputation by Last Observation Carried Forward
	nocb	Imputation by Next Observation Carried Backward
na.ma		
	simple	Missing Value Imputation by Simple Moving Average
	linear	Missing Value Imputation by Linear Weighted Moving Average
	exponential	Missing Value Imputation by Exponential Weighted Moving Average
na.mean		
	mean	MissingValue Imputation by Mean Value
	median	Missing Value Imputation by Median Value
	mode	Missing Value Imputation by Mode Value
na.random		Missing Value Imputation by Random Sample
na.replace		Replace Missing Values by a Defined Value

Summary and Q&A

- **Motivation and Terminology**
- **Data Cleaning and Fusion**
- **Missing Value Imputation**

- **Next Lectures (Part B)**
  - **08 Cloud Computing Foundations** [Nov 17]