Foundation of data engineering [IF-5-OT7:TD]

MCF Riccardo Tommasini

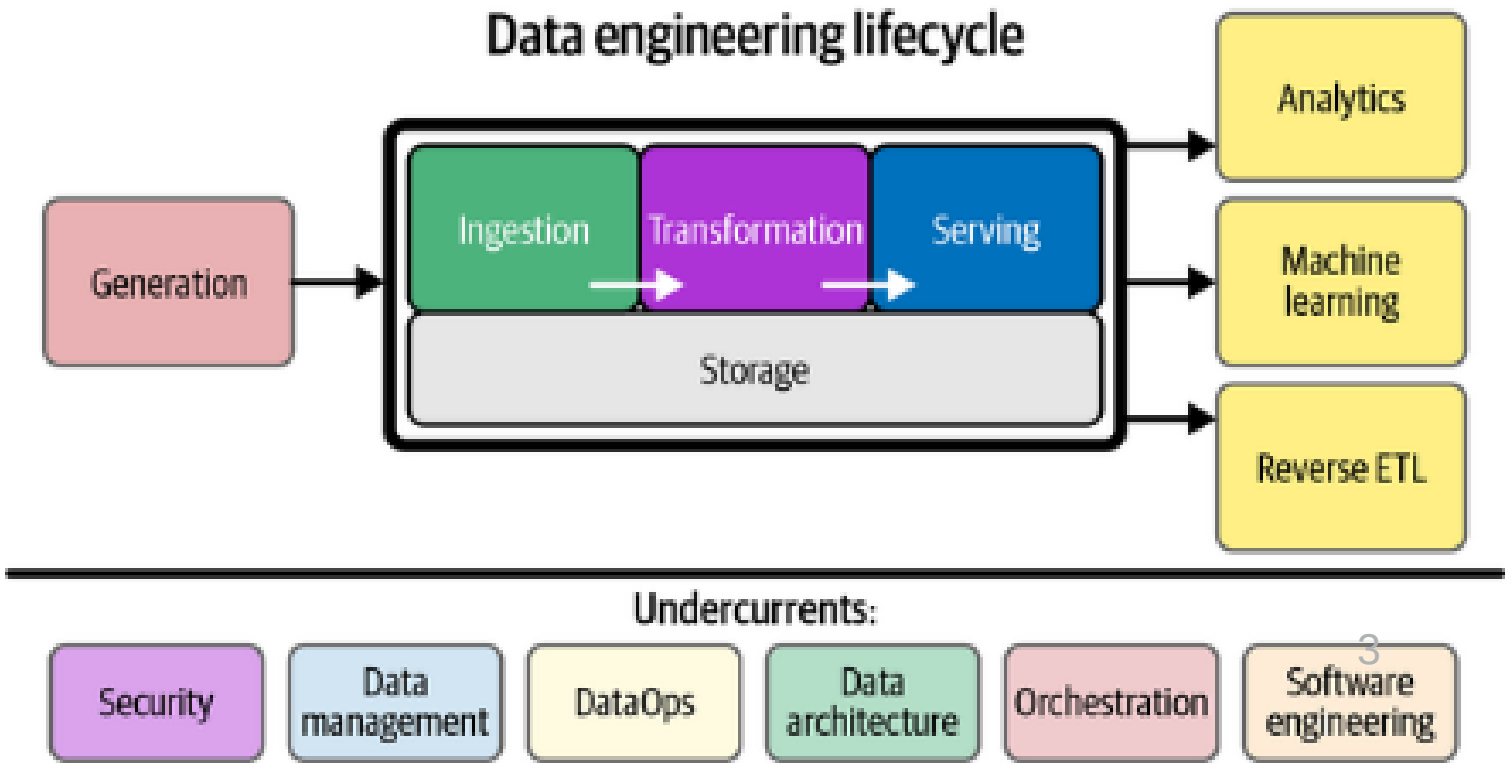
http://rictomm.me

riccardo.tommasini@insa-lyon.fr

Data Transformation (Wrangling)

Slides by Kristo Raun

Data (Engineering) Lifecycle



Agenda

1. What is data wrangling?
2. Why is data wrangling necessary?
3. Types of data wrangling
4. Levels of data wrangling
5. Considerations around data wrangling

What is data wrangling?

What is data wrangling?

Data cleaning

What is data wrangling?

Data cleaning

Data cleansing

What is data wrangling?

Data cleaning

Data cleansing

Data munging

What is data wrangling?

Data cleaning

Data cleansing

Data munging

Data preprocessing

What is data wrangling?

Data cleaning

Data cleansing

Data munging

Data preprocessing

Data preparation

What is data wrangling?

Data cleaning

Data cleansing

Data munging

Data preprocessing

Data preparation

Data mapping

What is data wrangling?

Data cleaning

Data cleansing

Data munging

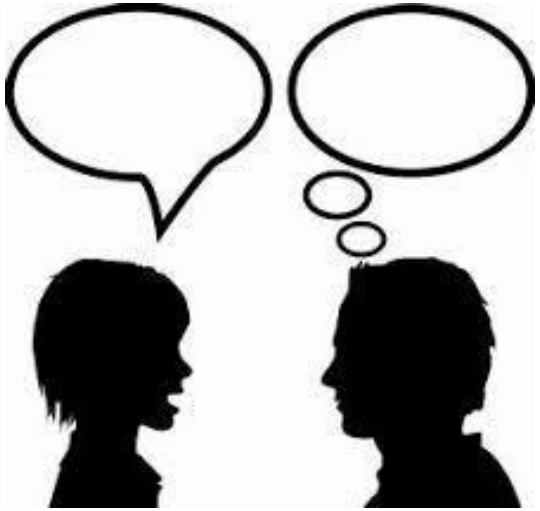
Data preprocessing

Data preparation

Data mapping

Data transformation

What is data wrangling?



Data cleaning

Data cleansing

Data munging

Data preprocessing

Data preparation

Data mapping

Data transformation

What is data wrangling?

What is data wrangling?

We need data in the form of Z

What is data wrangling?

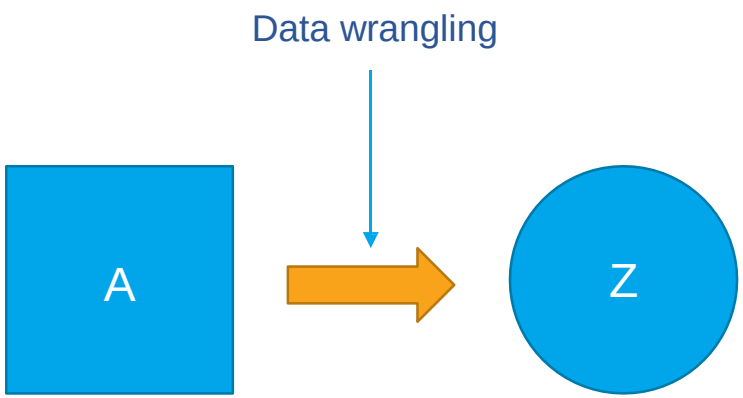
We need data in the form of Z

We have data in the form of A

What is data wrangling?

We need data in the form of Z

We have data in the form of A



What is data wrangling?

We want to…

- **Change data format**

27.09.2021 ➔ 2021-09-27

What is data wrangling?

We want to…

- Change data format
- **Change data type**

"500" ➔ 500

String ➔ Integer

What is data wrangling?

Name	Age
Andres	33
Anna	44
Augustus	?

We want to…

- Change data format
- Change data type
- **Fix missing values**

What is data wrangling?

Name	Age
Andres	33
Anna	*44*
Anna	*44*
Augustus	

We want to…

- Change data format
- Change data type
- Fix missing values
- **Fix duplicates**

What is data wrangling?

Name
Andres
Anna
Augustus

Name
Mr. Andres
Ms. Anna
Dr. Augustus

BI

ML

We want to…

- Change data format
- Change data type
- Fix missing values
- Fix duplicates
- **Augment**

What is data wrangling?

We want to…

- Change data format
- Change data type
- Fix missing values
- Fix duplicates
- Augment
- **Group**

What is data wrangling?

We want to…

- Change data format
- Change data type
- Fix missing values
- Fix duplicates
- Augment
- Group
- **Aggregate**

What is data wrangling?

Name	Age
Andres	33
Anna	44

Where age < 40

Name	Age
Andres	33

We want to…

- Change data format
- Change data type
- Fix missing values
- Fix duplicates
- Augment
- Group
- Aggregate
- **Filter**

What is data wrangling?

Id	Name
1	Andres
2	Anna

OrderNumber	CustomerId
2021_A	1
2021_B	2
2021_C	2

OrderNumber	CustomerName
2021_A	Andres
2021_B	Anna
2021_C	Anna

We want to…

- Change data format
- Change data type
- Fix missing values
- Fix duplicates
- Augment
- Group
- Aggregate
- Filter
- **Join**

What is data wrangling?

We want to…

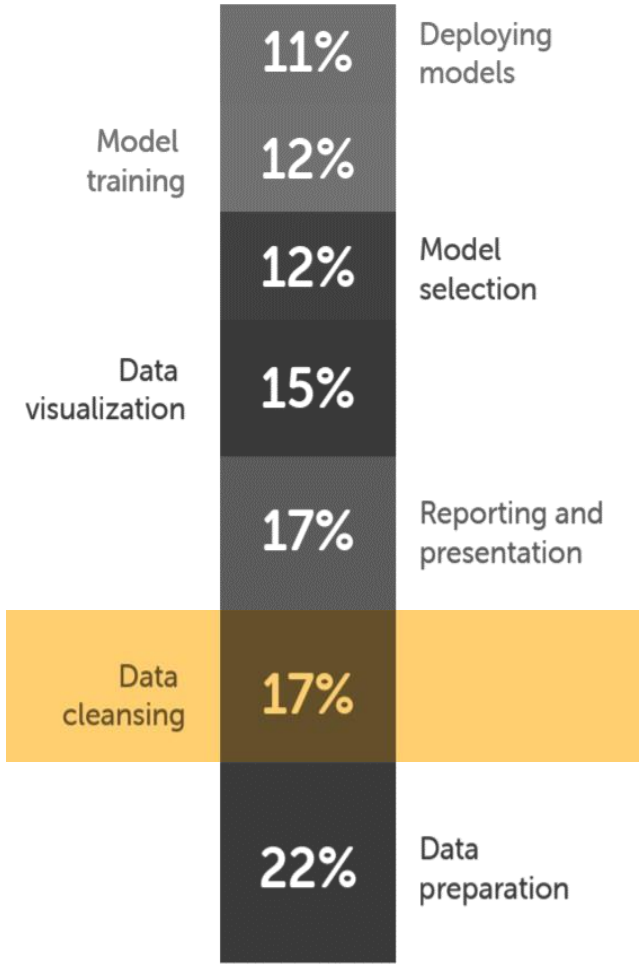
- Change data format
- Change data type
- Fix missing values
- Fix duplicates
- Augment
- Group
- Aggregate
- Filter
- Join

Why is data wrangling necessary?

Why is data wrangling necessary?

Why is data wrangling necessary?

How do data scientists spend their time?

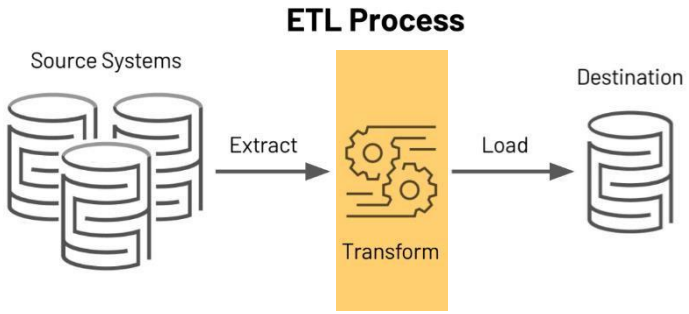


Source: 2021 State of Data Science Report
Anaconda
N = 2 030

The ML view:

- Models assume data in certain format
- Models should have as clean data as possible

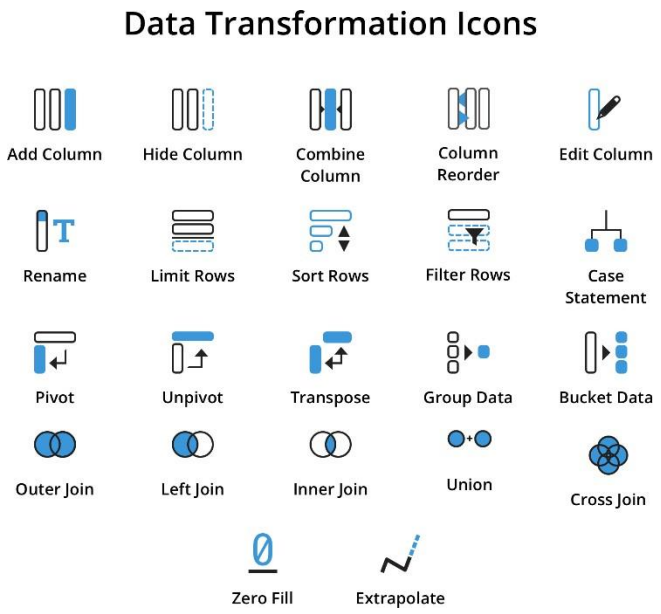
Why is data wrangling necessary?



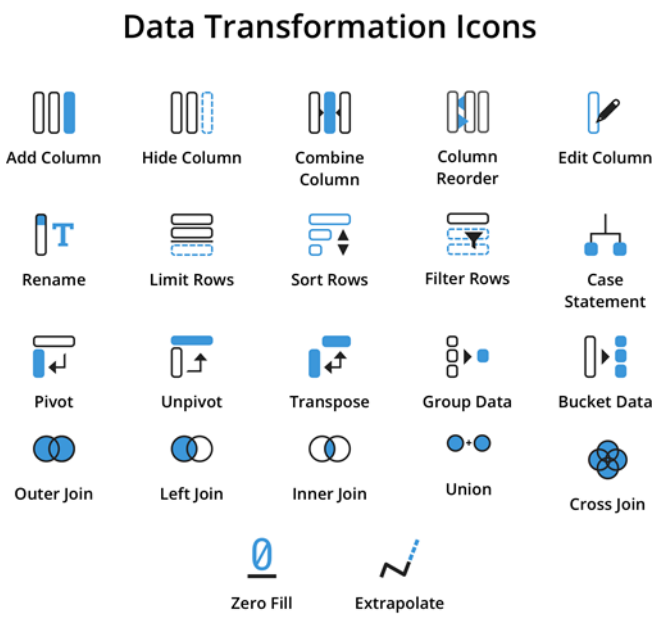
The BI view:

- Data ➔ Insights ➔ Actions.
  - Data needs to be usable
  - Data needs to be trustworthy
  - Data needs to be available

Types of data wrangling

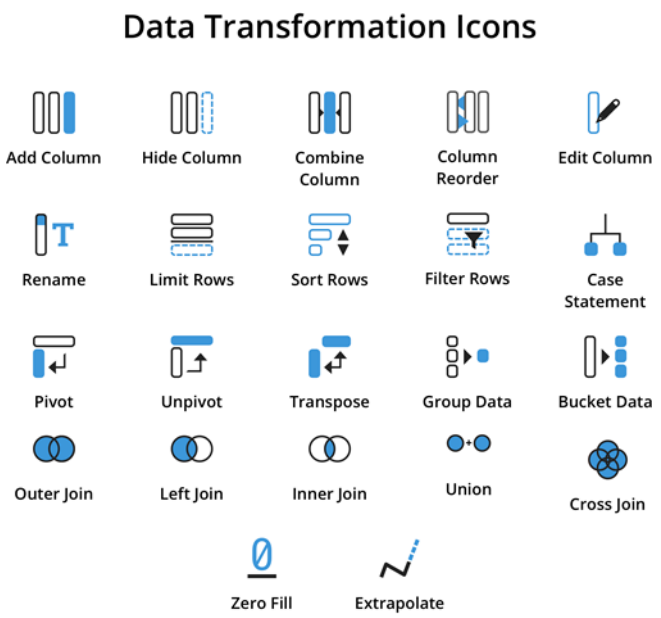


Types of data wrangling



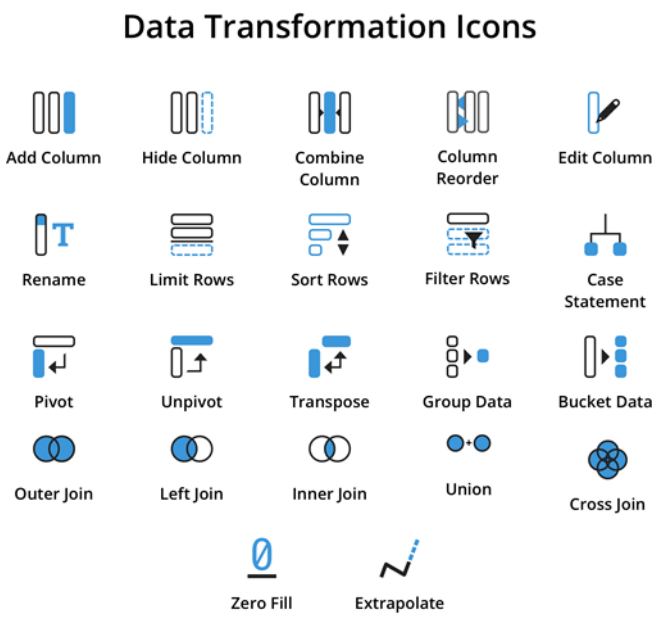
- **Validity of values**
  - Phone numbers
    - “+372 51234567”
    - “00372 51234567”
    - “512 345 67”
    - 51234567
  - ID codes and references
    - PID (EE)
      - Gender
      - Date of birth
      - Checksum
    - Reference Number of the Invoice (EE)

Types of data wrangling



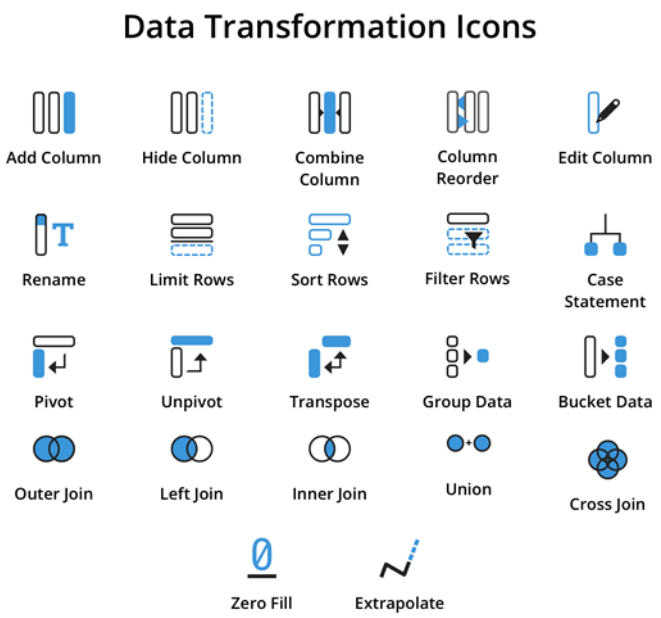
- **Consistent values**
  - Does zip code + city make sense?
  - Can there be a sales order worth more than 1 million euros?
  - If a customer has conflicting e-mails in different systems, which system is correct?

Types of data wrangling



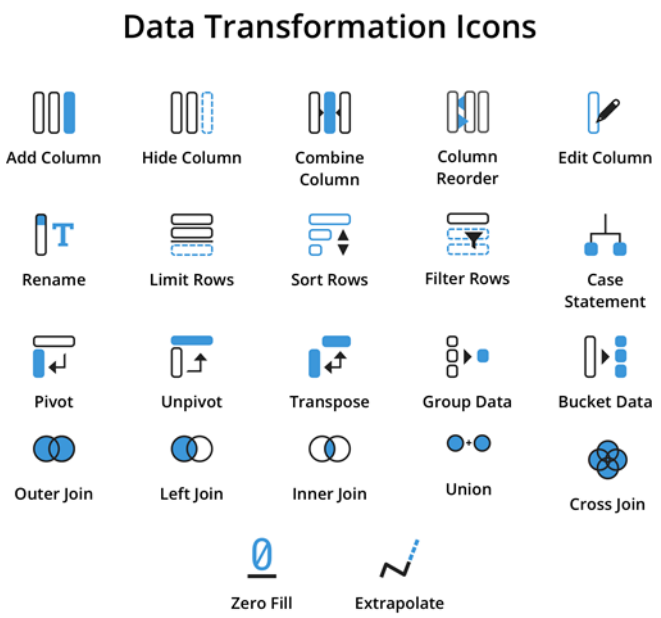
- **Duplicates**
  - Do we accept duplicates?
  - Is it possible to set validity of data (updated timestamp)?
  - Keep only one row (try to make the choice idempotent)
  - Can we trust the source system to have unique values?

Types of data wrangling



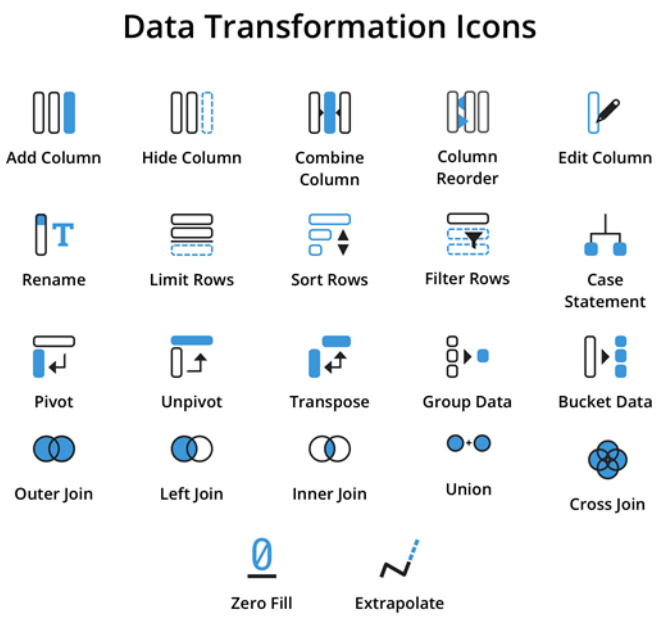
- **Business rules**
  - Does a premium customer have the associated premium services?
  - Is this product allowed to have this discount?

Types of data wrangling



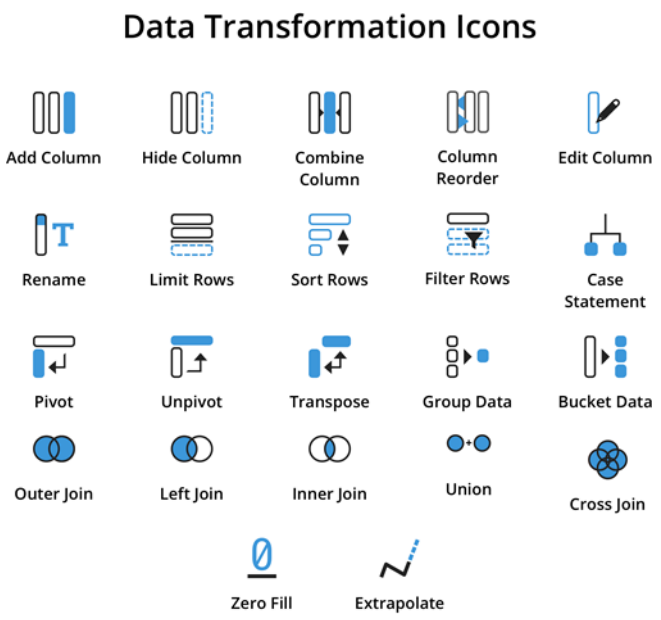
- **Conforming values**
  - If customer has conflicting emails in different systems, which system is correct?
  - Is there a unique code across systems for defining a … (product, customer, location, …)
    - Eg product name in ERP vs sales system vs website?

Types of data wrangling



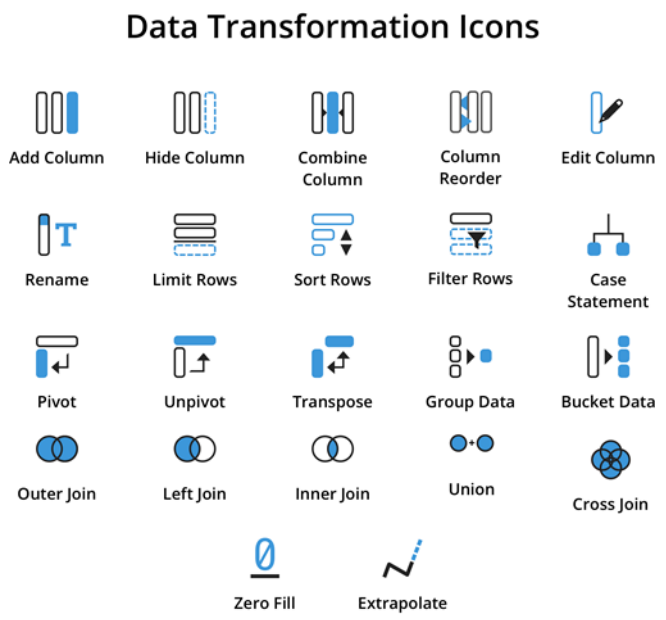
- **Missing data**
  - What is NULL?
  - Is NULL acceptable?
    - Aggregation over NULL is (usually) correct
    - -1 (or similar) to use for referential integrity
  - In data science/ML:
    - Delete data
    - Impute data
      - Easiest/fastest: median

Types of data wrangling



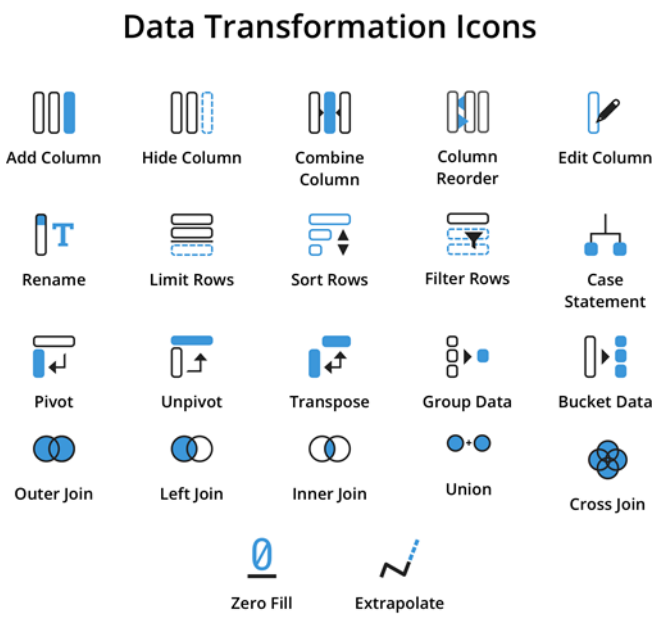
- **Wrong data type**
  - Schema definition
    - Load in csv without schema - everything is a string
    - Inferring schema (eg Spark) can end up with wrong type
  - Unit
    - String vs integer vs decimal type
      - A hundred pieces
      - 100 pieces
      - 100.55 pieces
  - Timestamps
    - UTC vs local
    - UNIX timestamp
      - Avro files – date/timestamp is integer. Eg how many days since 1 January 1970 (ISO calendar)

Types of data wrangling



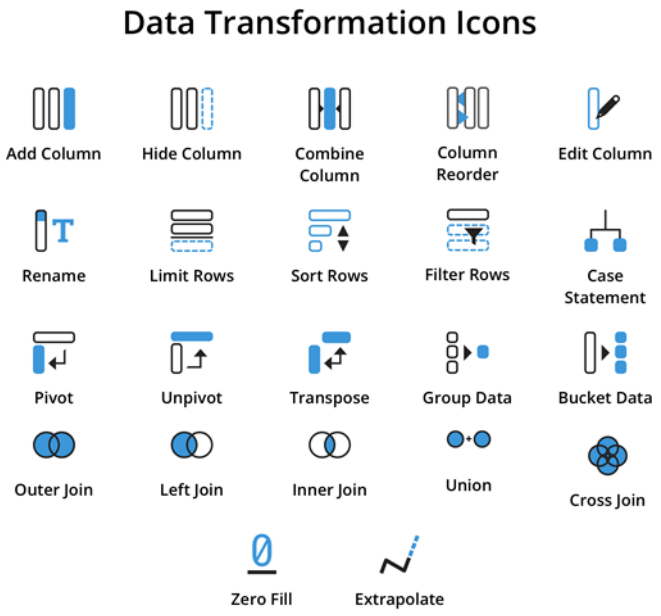
- **Wrong data structure**
  - Structured data
    - CSV
    - Excel
    - Conventional database
  - Nested data (semistructured data)
    - JSON
    - Parquet
    - Struct, array (in Spark, BigQuery, etc)
  - Unstructured data
    - Text
    - Images
    - Audio/Video

Types of data wrangling



- **Aggregations**
  - Grouping
    - Correct grouping columns (level of slice/dice)
  - How will the data be used (visualization, reporting, ML)

Types of data wrangling



- Validity of values
- Consistent values
- Duplicates
- Business rules
- Conforming values
- Missing data
- Wrong data type
- Wrong data structure
- Aggregations
- …

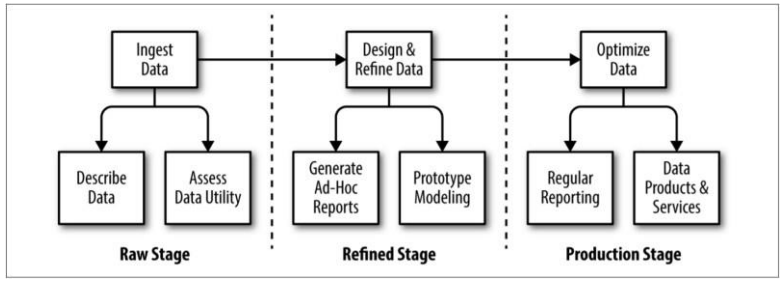
Levels of data wrangling

Levels of data wrangling

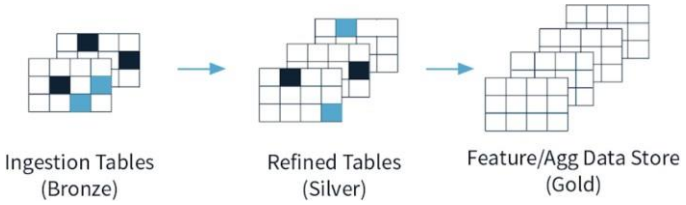
Abstractly – how far are you in the data wrangling process?

Various definitions, commonly 3 stages are defined

Levels of data wrangling



Source: Trifacta

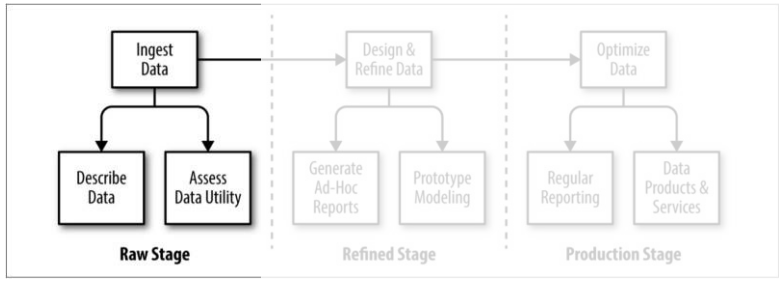


Source: Databricks

Abstractly – how far are you in the data wrangling process?

Various definitions, commonly 3 stages are defined

Levels of data wrangling

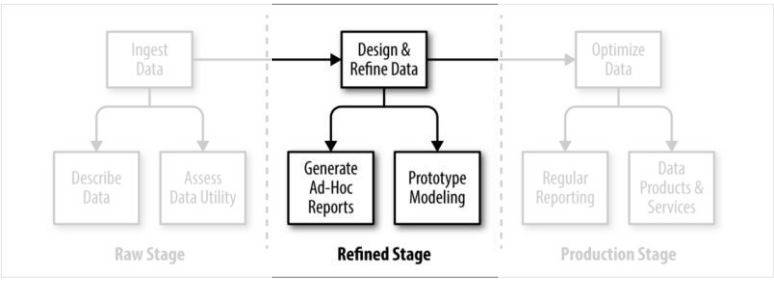


Source: Trifacta

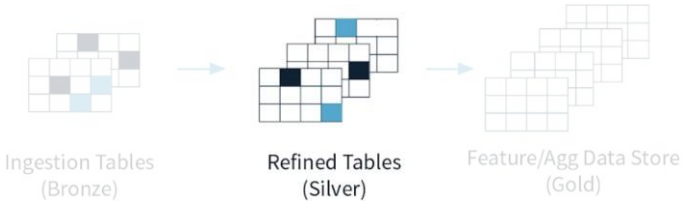
Source: Databricks

OrderNumber	OrderRevenue	OrderTimestamp	CustomerId
O_12345	150.00	2021-09-01T13:00:05+03:00	55
O_12346	230.00	2021-09-01T13:20:11+03:00	78
O_12347	170.00	2021-09-01T12:55:22+02:00	41
R_12346	230.00	2021-09-01T13:56:05+03:00	78
O_12348	50.50	2021-09-01T14:01:05+03:00	97
O_12349	450.23	2021-09-01T15:12:05+03:00	55

Levels of data wrangling



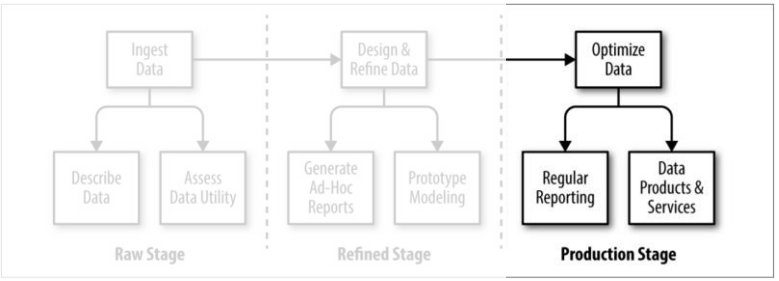
Source: Trifacta



Source: Databricks

OrderNumber	OrderRevenue	OrderTimestampUTC	CustomerId	OrderType	IsReturned
O_12345	150.00	2021-09-01 10:00:05	55	100	0
O_12346	230.00	2021-09-01 10:20:11	78	100	1
O_12347	170.00	2021-09-01 10:55:22	41	100	0
R_12346	-230.00	2021-09-01 10:56:05	78	400	NULL
O_12348	50.50	2021-09-01 11:01:05	97	100	0
O_12349	450.23	2021-09-01 12:12:05	55	100	0

Levels of data wrangling



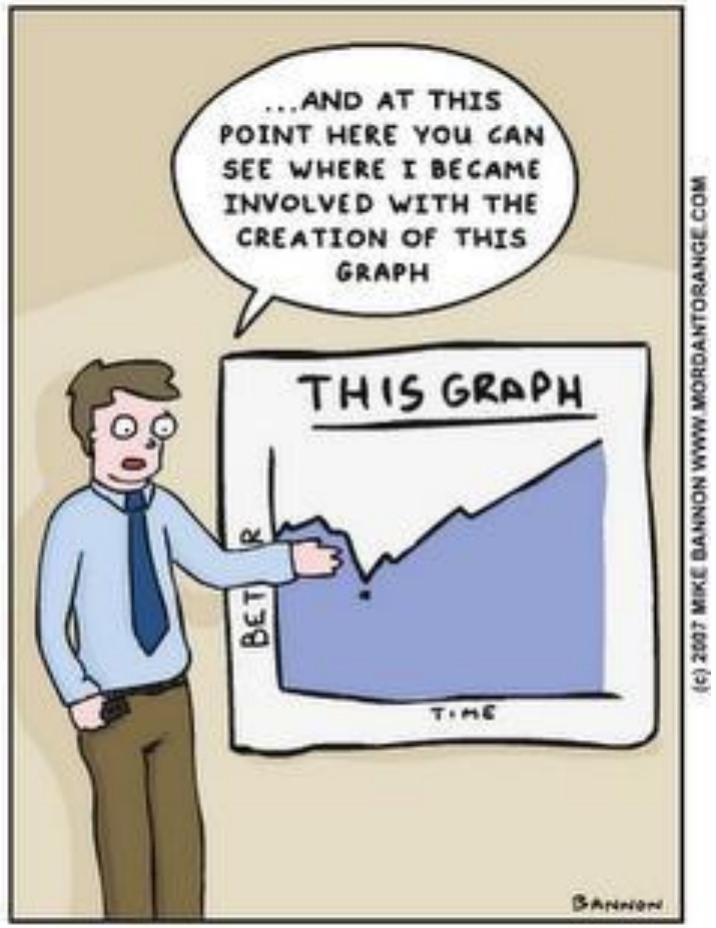
Source: Trifacta

Source: Databricks

OrderNumber	OrderRevenue	OrderTimestampUTC	OrderDate	CustomerId	OrderType	IsReturned
O_12345	150.00	2021-09-01 10:00:05	2021-09-01	55	100	0
O_12347	170.00	2021-09-01 10:55:22	2021-09-01	41	100	0
O_12348	50.50	2021-09-01 11:01:05	2021-09-01	97	100	0
O_12349	450.23	2021-09-01 12:12:05	2021-09-01	55	100	0

Considerations

Data wrangling approaches

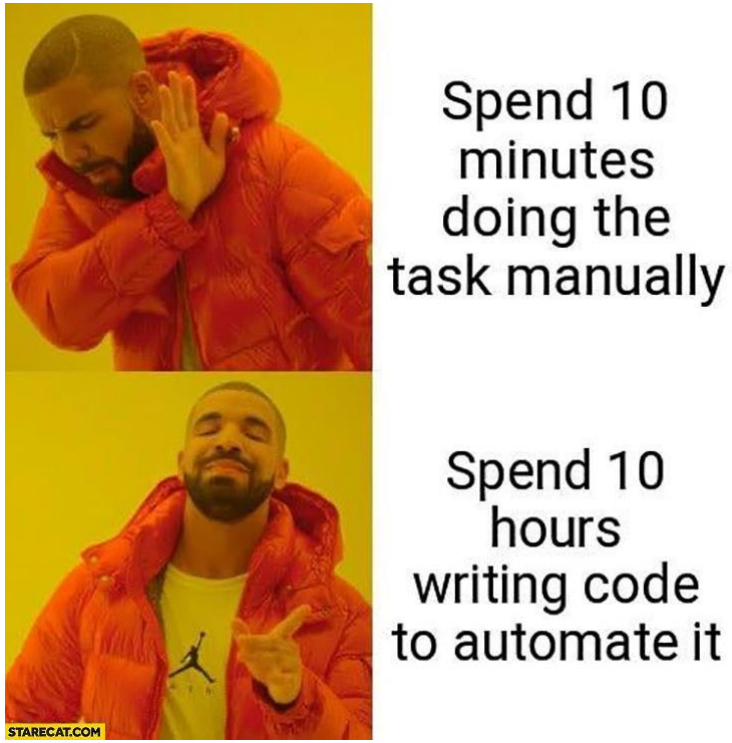


Considerations

Data wrangling approaches

- Adhoc / exploratory / PoC

Considerations



Data wrangling approaches

- Adhoc / exploratory / PoC
- Production data engineering

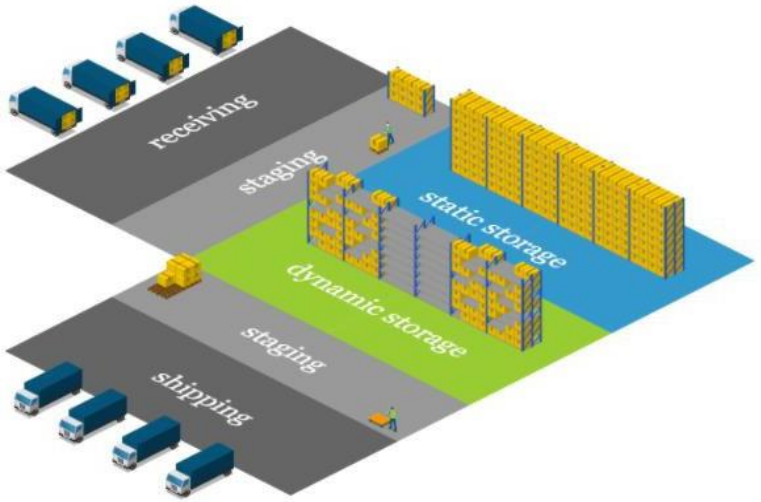
Considerations

Data wrangling approaches

- Adhoc / exploratory / PoC
- Production data engineering

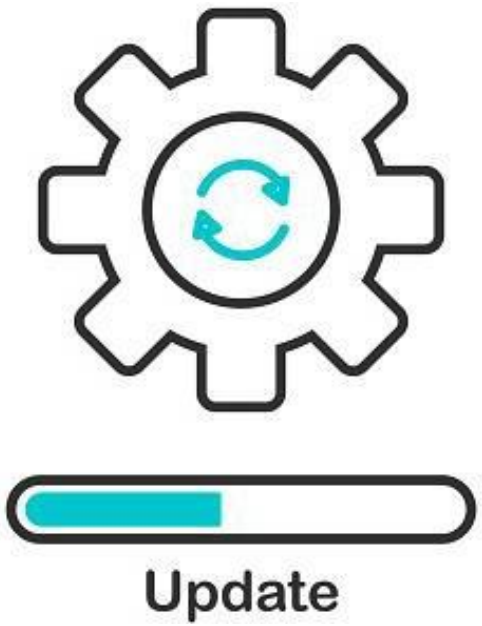
Consider which tool to use for which purpose

Considerations



- Staging area
  - *Temporary* place for data
    - Raw data from source(s)
    - Transformation steps
  - Important for following ELT (instead of ETL)
    - Storage is cheap, compute is expensive
    - Minimize impact on source systems
  - Detect changes

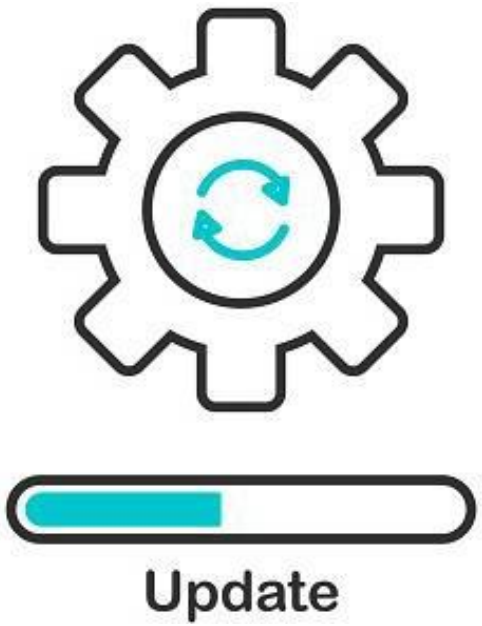
Considerations



- What happens if source data is updated?

Customers *Orders*

Considerations



- What happens if source data is updated?

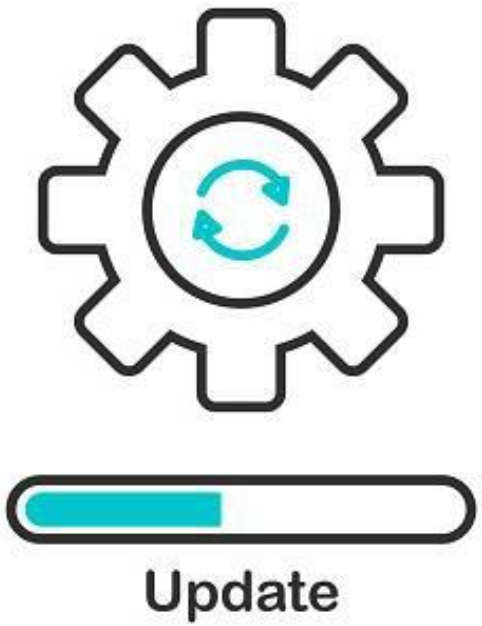
Customers

CustomerId	Zip code
5	10140

Orders

CustomerId	OrderId	OrderRevenue
5	20210901_001	2 000.00

Considerations



- What happens if source data is updated?

Customers

CustomerId	Zip code
5	10140

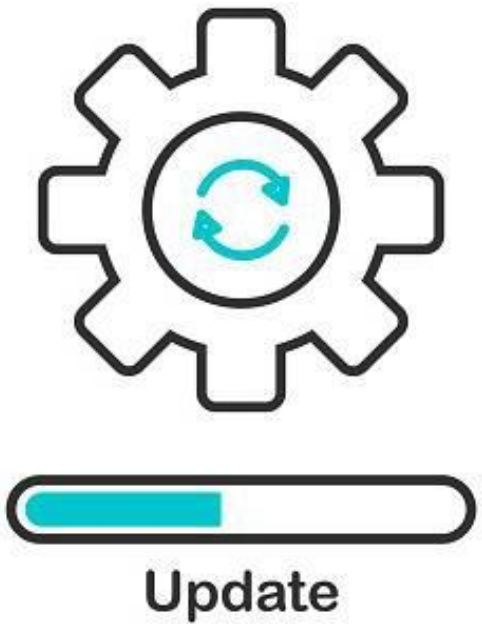
CustomerId	Zip code
5	51009

Orders

CustomerId	OrderId	OrderRevenue
5	20210901_001	2 000.00

CustomerId	OrderId	OrderRevenue
5	20210901_001	2 000.00
5	20210902_001	4 500.00

Considerations



- What happens if source data is updated?

Customers

CustomerId	Zip code
5	10140

CustomerId	Zip code
5	51009

CustomerId	Zip code
5	90210

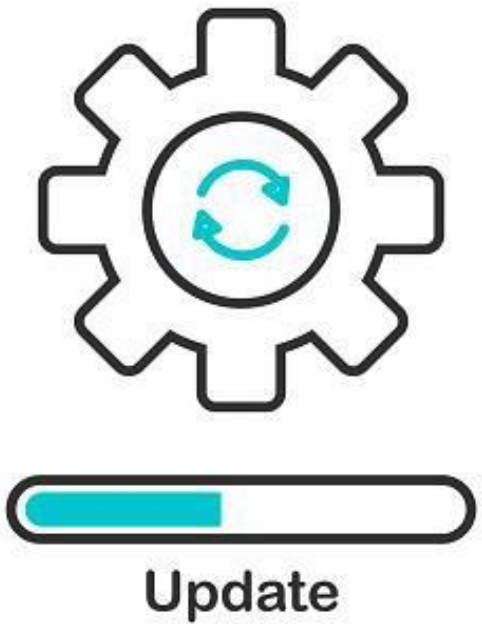
Orders

CustomerId	OrderId	OrderRevenue
5	20210901_001	2 000.00

CustomerId	OrderId	OrderRevenue
5	20210901_001	2 000.00
5	20210902_001	4 500.00

CustomerId	OrderId	OrderRevenue
5	20210901_001	2 000.00
5	20210902_001	4 500.00
5	20210903_001	300.00

Considerations



- What happens if source data is updated?

Customers

Id	CustomerId	Zip code	ValidFrom	ValidTo
1	5	10140	2021-09-01	9999-12-31

Id	CustomerId	Zip code	ValidFrom	ValidTo
1	5	10140	2021-09-01	2021-09-01
2	5	51009	2021-09-02	9999-12-31

Id	CustomerId	Zip code	ValidFrom	ValidTo
1	5	10140	2021-09-01	2021-09-01
2	5	51009	2021-09-02	2021-09-02
3	5	90210	2021-09-033	9999-12-31

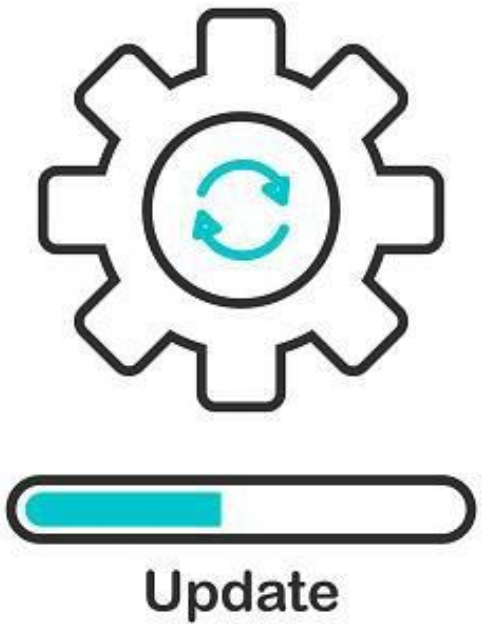
Orders

CustomerId	OrderId	OrderRevenue
1	20210901_001	2 000.00

CustomerId	OrderId	OrderRevenue
1	20210901_001	2 000.00
2	20210902_001	4 500.00

CustomerId	OrderId	OrderRevenue
1	20210901_001	2 000.00
2	20210902_001	4 500.00
3	20210903_001	300.00

Considerations



- What happens if source data is updated?
  - Slowly changing dimensions
    - Type 0
      - Always original – e.g. date of birth
    - Type 1
      - Always overwrite – simple but no history (often incorrect)
    - Type 2
      - New row – good for history, becomes complex
    - Other types:
      - history tables, history attributes, combinations

Considerations

- How is data published to target?
  - Fact and dimension tables
  - ML model
  - BI data files
  - Kafka
  - API
  - …?

Considerations

- Data wrangling is *usually* not a siloed activity. You need to consider
  - Integration
    - Where is data coming from? Which sources have the same data? Which source holds the master data?
  - Infrastructure
    - Which systems hold the data? Which format is source data in? Which system is used for transformation? On premise vs cloud vs hybrid infrastructure?
  - Security
    - Who has/should have access to source data? Who should have access to data wrangling code?
  - Exception handling
    - What is the system criticality? Is it better to allow data wrangling to fail or skip/ignore? Logging and notification of errors.
  - Data Quality
    - Are there contracts in place towards data source? What types of data quality errors are allowed in the output data?
  - Data modelling
    - Kimball vs Inmon vs Data vault – which mindset/framework are we using? Do we have a lot of streaming inserts? Do we have a lot of ad hoc querys?

Considerations

- Where does data wrangling *start* from?
  - Feedback loop to source system
  - Issues arising from UI / user forms in source systems
    - Web sites, Internal systems
    - Enforcing data validations vs UX


pandas

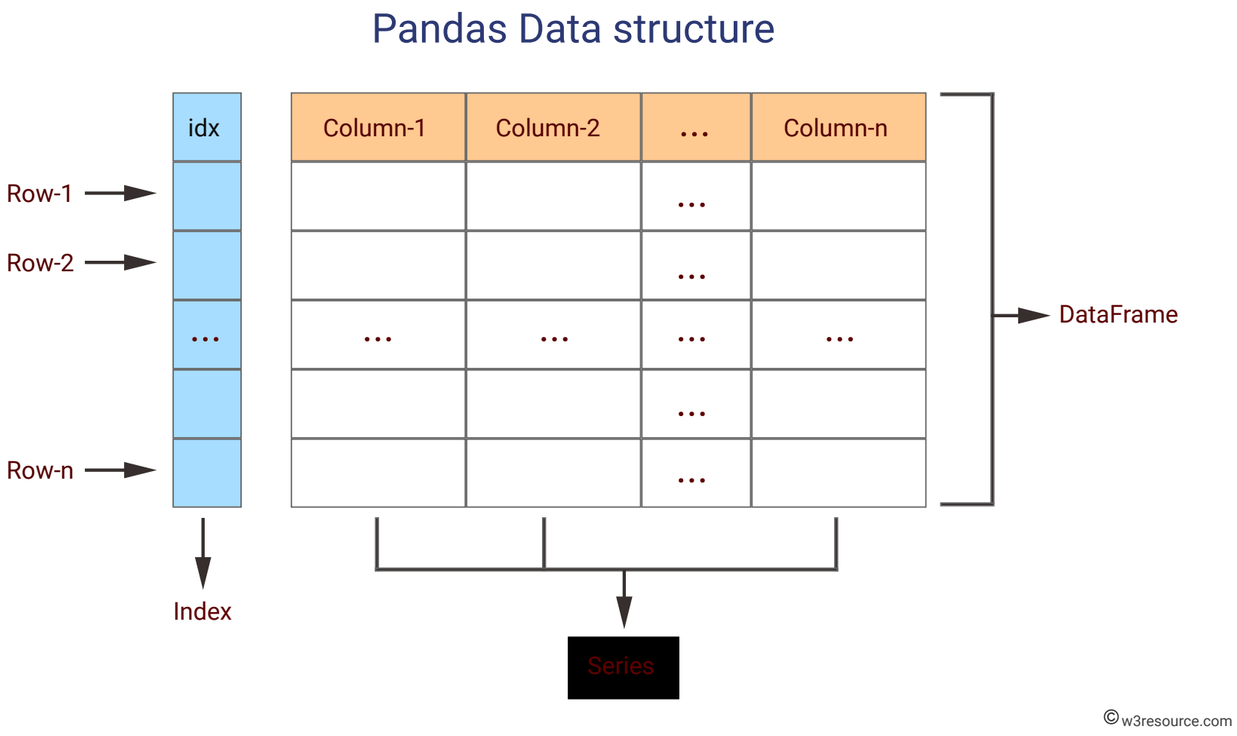
Abstract

- Pandas is one of the most popular data analysis tools
- It is open-source, written in python/C++
- It is flexible, powerful, fast, and quite easy-to-use.
- Pandas covers a number of use cases
- Data Analysis - What are the data about?
- Data Transformation - How do the data need to look like?

DataFrames

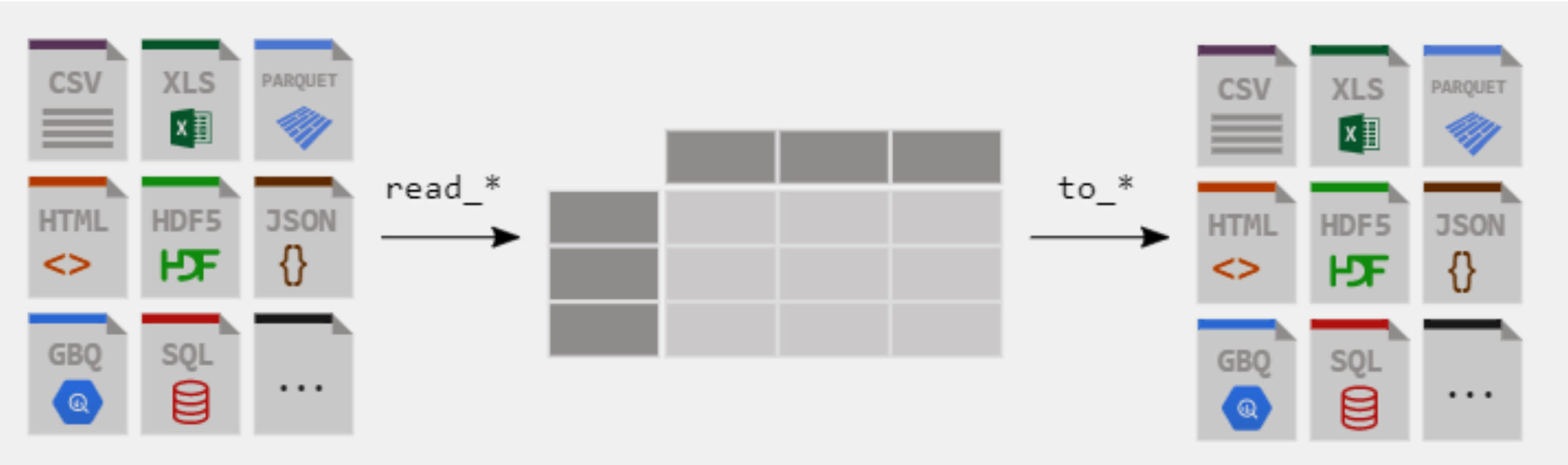
A DataFrame is **a data structure that organises data into a 2-dimensional table of rows and columns**, much like a spreadsheet.
DataFrames are one of the most common data structures used in modern data analytics because they are a flexible and intuitive way of storing and working with data.

— *Databricks*



DataFrame (cont.)

- They can handle any sort of tabular data
- They support (and can normalise) nested data



Data Wrangling
with pandas
Cheat Sheet
http://pandas.pydata.org

Tidy Data – A foundation for wrangling in pandas

In a tidy data set:

Each **variable** is saved in its own **column**

&

Each **observation** is saved in its own **row**

Tidy data complements pandas's vectorized operations. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.

M * A

Syntax – Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
        {"a" : [4 ,5, 6],
         "b" : [7, 8, 9],
         "c" : [10, 11, 12]},
       index = [1, 2, 3])
```

Specify values for each column.

```
df = pd.DataFrame(
     [[4, 7, 10],
      [5, 8, 11],
      [6, 9, 12]],
     index=[1, 2, 3],
     columns=['a', 'b', 'c'])
```

Specify values for each row.

		a	b	c
n	v			
d	1	4	7	10
	2	5	8	11
e	2	6	9	12

```
df = pd.DataFrame(
        {"a" : [4 ,5, 6],
         "b" : [7, 8, 9],
         "c" : [10, 11, 12]},
index = pd.MultiIndex.from_tuples(
        [('d',1),('d',2),('e',2)],
          names=['n','v']))
```

Create DataFrame with a MultiIndex

Method Chaining

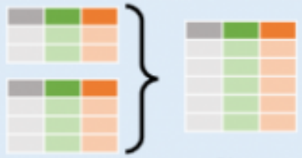
Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
        .rename(columns={
                'variable' : 'var',
                'value' : 'val'})
        .query('val >= 200')
     )
```

Reshaping Data – Change the layout of a data set

pd.melt(df)
Gather columns into rows.

df.pivot(columns='var', values='val')
Spread rows into columns.



pd.concat([df1,df2])
Append rows of DataFrames

pd.concat([df1,df2], axis=1)
Append columns of DataFrames

df.sort_values('mpg')
Order rows by values of a column (low to high).

df.sort_values('mpg',ascending=False)
Order rows by values of a column (high to low).

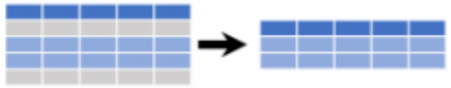
df.rename(columns = {'y':'year'})
Rename the columns of a DataFrame

df.sort_index()
Sort the index of a DataFrame

df.reset_index()
Reset index of DataFrame to row numbers, moving index to columns.

df.drop(columns=['Length','Height'])
Drop columns from DataFrame

Subset Observations (Rows)



df[df.Length > 7]
Extract rows that meet logical criteria.

df.drop_duplicates()
Remove duplicate rows (only considers columns).

df.head(n)
Select first n rows.

df.tail(n)
Select last n rows.

df.sample(frac=0.5)
Randomly select fraction of rows.

df.sample(n=10)
Randomly select n rows.

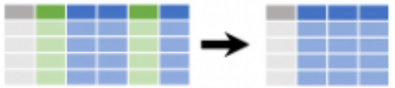
df.iloc[10:20]
Select rows by position.

df.nlargest(n, 'value')
Select and order top n entries.

df.nsmallest(n, 'value')
Select and order bottom n entries.

Logic in Python (and pandas)			
<	Less than	!=	Not equal to
>	Greater than	df.column.isin(*values*)	Group membership
==	Equals	pd.isnull(*obj*)	Is NaN
<=	Less than or equals	pd.notnull(*obj*)	Is not NaN
>=	Greater than or equals	&,|,~,^,df.any(),df.all()	Logical and, or, not, xor, any, all

Subset Variables (Columns)



df[['width','length','species']]
Select multiple columns with specific names.

df['width'] *or* **df.width**
Select single column with specific name.

df.filter(regex='*regex*')
Select columns whose name matches regular expression *regex*.

regex (Regular Expressions) Examples	
'\.'	Matches strings containing a period '.'
'Length$'	Matches strings ending with word 'Length'
'^Sepal'	Matches strings beginning with the word 'Sepal'
'^x[1-5]$'	Matches strings beginning with 'x' and ending with 1,2,3,4,5
'^(?!Species$).*'	Matches strings except the string 'Species'

df.loc[:,'x2':'x4']
Select all columns between x2 and x4 (inclusive).

df.iloc[:,[1,2,5]]
Select columns in positions 1, 2 and 5 (first column is 0).

df.loc[df['a'] > 10, ['a','c']]
Select rows meeting logical condition, and only the specific columns .