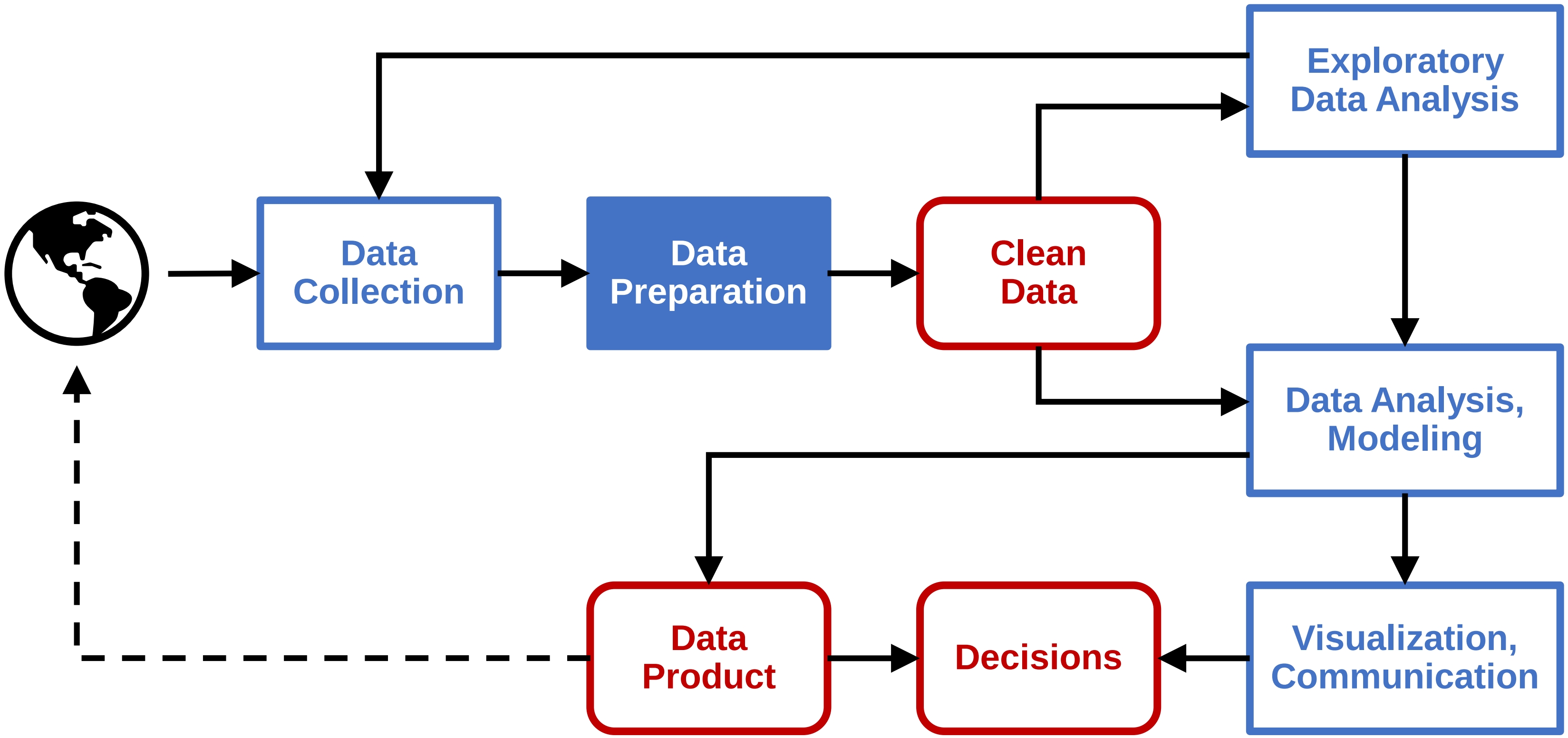
Data Preparation

CS 418. Introduction to Data Science

Review
The Data Science Process

- The goal of **data science process** is to **extract knowledge or insights from data**.



Adapted from: Cathy O'Neil and Rachel Schutt, *Doing Data Science* (2013)

Data Preparation
Introduction

"It is often said that 80% of data analysis is spent on the process of cleaning and preparing the data."

- Raw data is often messy.
- The purpose of **data preparation** is to:
  - Detect and correct **data quality** problems.
    - Presence of **noise and outliers**.
    - **Missing**, **inconsistent**, or **duplicate data**.
  - Transform the raw data into an appropriate format for data analysis.
    - **Aggregation**.
    - **Sampling**.
    - **Variable transformation**.
    - **Dimensionality reduction**.
- Also called **data preprocessing**, **data cleaning**, **data wrangling**, or **data munging**.

Data Preparation
Exercise 5.1

THINK
PAIR
SHARE

Each of the following tables shows the same data organized in different ways. Which one would you choose to facilitate data analysis?

Option A

	country	year	type	count
	<chr>	<int>	<chr>	<int>
1	Afghanistan	1999	cases	745
2	Afghanistan	1999	population	19987071
3	Afghanistan	2000	cases	2666
4	Afghanistan	2000	population	20595360
5	Brazil	1999	cases	37737
6	Brazil	1999	population	172006362

... with 6 more rows

Option B

	country	year	cases	population
	<chr>	<int>	<int>	<int>
1	Afghanistan	1999	745	19987071
2	Afghanistan	2000	2666	20595360
3	Brazil	1999	37737	172006362
4	Brazil	2000	80488	174504898
5	China	1999	212258	1272915272
6	China	2000	213766	1280428583

Option C

	country	year	rate
*	<chr>	<int>	<chr>
1	Afghanistan	1999	745/19987071
2	Afghanistan	2000	2666/20595360
3	Brazil	1999	37737/172006362
4	Brazil	2000	80488/174504898
5	China	1999	212258/1272915272
6	China	2000	213766/1280428583

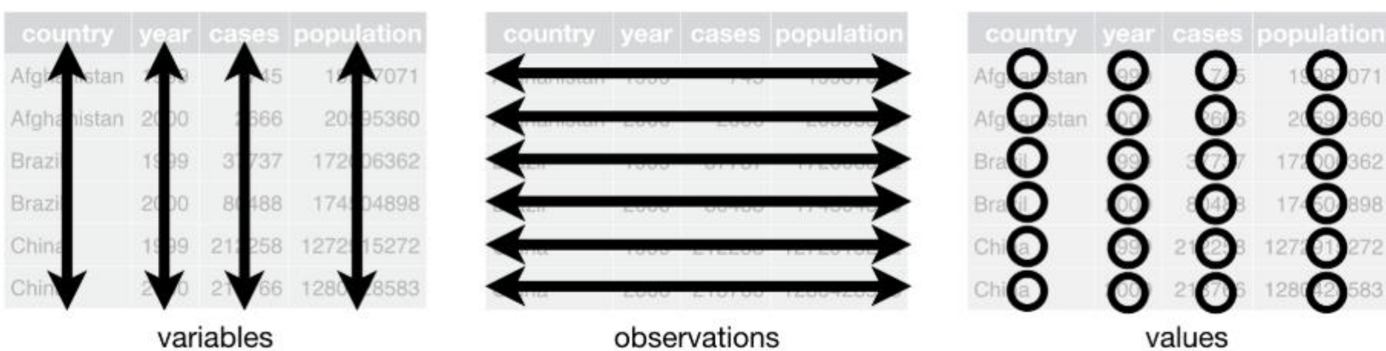
Option D

	country	`1999`	`2000`
*	<chr>	<int>	<int>
1	Afghanistan	745	2666
2	Brazil	37737	80488
3	China	212258	213766

	country	`1999`	`2000`
*	<chr>	<int>	<int>
1	Afghanistan	19987071	20595360
2	Brazil	172006362	174504898
3	China	1272915272	1280428583

Data Preparation
Tidy Data

- **Tidy data** is a framework to structure datasets so they can be easily analyzed and visualized.
  - Each **row** is an **observation**.
  - Each **column** is a **variable**.
  - Each **value** must have its own **cell**.
  - Each type of **observational unit** forms a **table**.



Source: Garrett Grolemund and Hadley Wickham, *R for Data Science* (2016)

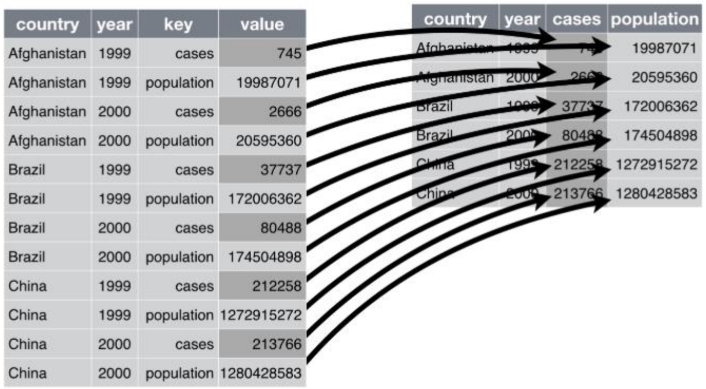
- Note that **not all data** can be structured as **tidy data**.

Tidy Data
Common Problems (I)

- Real datasets are often not structured as **tidy data**. Some common problems are:
  - **One observation is stored in multiple rows.**
  - ***Example*:**

	country	year	type	count
	<chr>	<int>	<chr>	<int>
1	Afghanistan	1999	cases	745
2	Afghanistan	1999	population	19987071
3	Afghanistan	2000	cases	2666
4	Afghanistan	2000	population	20595360

  - **How to fix?** Reshape dataset from long format to wide format.



In Python (with **pandas**):

```
import pandas as pd
pd.pivot_table(data,…)
```

Click for documentation

Tidy Data
Common Problems (II)

- Real datasets are often not structured as **tidy data**. Some common problems are:
  - **Column headers are values, not variable names.**
  - ***Example*:**

```
  country     `1999` `2000`
* <chr>        <int>  <int>
1 Afghanistan    745   2666
2 Brazil       37737  80488
3 China       212258 213766
```

  - **How to fix?** Reshape dataset from wide format to long format.

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

In Python (with **pandas**):

```python
import pandas as pd
pd.melt(data,…)
```

Click for documentation

Tidy Data
Common Problems (III)

- Real datasets are often not structured as **tidy data**. Some common problems are:
  - **Multiple variables are stored in one column.**
  - ***Example*:**

```
  country      year  rate
* <chr>       <int>  <chr>
1 Afghanistan  1999  745/19987071
2 Afghanistan  2000  2666/20595360
3 Brazil       1999  37737/172006362
4 Brazil       2000  80488/174504898
5 China        1999  212258/1272915272
```

  - **How to fix?** Split into two columns.

country	year	rate
Afghanistan	1999	**745** / 19987071
Afghanistan	2000	**2666** / 20595360
Brazil	1999	**37737** / 172006362
Brazil	2000	**80488** / 174504898
China	1999	**212258** / 1272915272
China	2000	**213766** / 1280428583

country	year	cases	population
Afghanistan	1999	**745**	19987071
Afghanistan	2000	**2666**	20595360
Brazil	1999	**37737**	172006362
Brazil	2000	**80488**	174504898
China	1999	**212258**	1272915272
China	2000	**213766**	1280428583

In Python (with **pandas**):

```
import pandas as pd
pd.Series.str.split(…)
```

Click for documentation

Tidy Data
Common Problems (IV)

- Real datasets are often not structured as **tidy data**. Some common problems are:
  - **A single observational unit is stored in multiple tables.**
  - ***Example*:**

	country	\`1999\`	\`2000\`
*	<chr>	<int>	<int>
1	Afghanistan	745	2666
2	Brazil	37737	80488
3	China	212258	213766

	country	\`1999\`	\`2000\`
*	<chr>	<int>	<int>
1	Afghanistan	19987071	20595360
2	Brazil	172006362	174504898
3	China	1272915272	1280428583

  - **How to fix?** Join datasets.

In Python (with **pandas**):

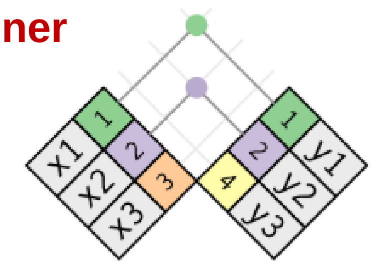
```
import pandas as pd
pd.DataFrame.merge(…)
```

Click for documentation

Tidy Data
Joining Datasets

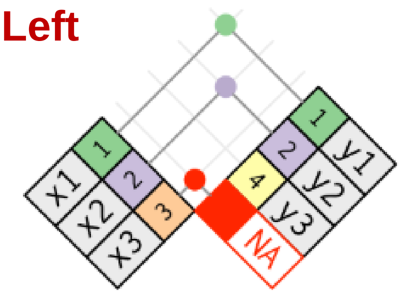
- A **join** is a way of connecting each row in the first (**left**) dataset to zero, one, or more rows in the second (**right**) dataset.

Inner



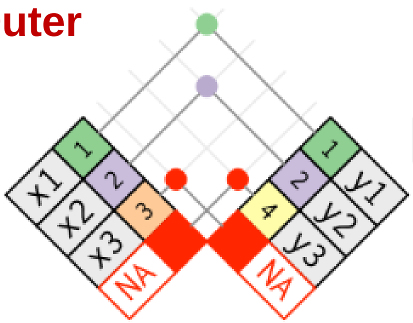
key	val_x	val_y
1	x1	y1
2	x2	y2

Left



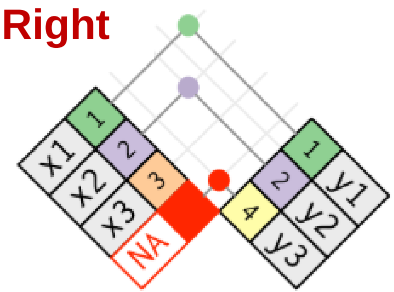
key	val_x	val_y
1	x1	y1
2	x2	y2
3	x3	NA

Outer



key	val_x	val_y
1	x1	y1
2	x2	y2
3	x3	NA
4	NA	y3

Right



key	val_x	val_y
1	x1	y1
2	x2	y2
4	NA	y3

Source: Garrett Grolemund and Hadley Wickham, [*R for Data Science*]() (2016)

Data Preparation
Outliers (I)

- An **outlier** is an observation that is very different from the rest of the observations in the dataset.
- There is not a standard definition of an **outlier**.
  - An **outlier** is an observation that is **greater than $Q_3 + 1.5(IQR)$ or less than $Q_1 - 1.5(IQR)$**.
    - The **interquartile range** (IQR) of a dataset is the difference between the third and the first quartile (**$Q_3 - Q_1$**).
  - An **outlier** is an observation that is **2 or 3 standard deviations away from the mean**.
- ***Example***:

3	37	23	61	36	65	6	24	1	19	92	1	13	40	1

$\boldsymbol{Q_3 + 1.5(IQR) = 89.5}$

$\boldsymbol{\bar{x} + 2s = 82.87}$

$\boldsymbol{\bar{x} + 3s = 110.24}$

Does the dataset contain any outliers?

Data Preparation
Outliers (II)

- Unlike **noise**, **outliers** can be valid observations.
- Domain knowledge is often required to determine if an **outlier** is an error or just an extreme but valid observation.
- Finding **outliers** is the goal of some data mining tasks (**anomaly detection**).
- **How to deal with outliers?**
  - **Outliers** can disproportionately influence models.
  - One approach is to create **two models** of the data: one with **outliers** and one without **outliers**.
  - If the conclusions are the same, the **outliers** can remain in the data.
  - Otherwise, further effort is needed to identify the cause of the **outliers**.

Data Preparation
Missing Values

- Real datasets often have **missing values**.
- **Missing values** can be:
  - **Explicit:** `NULL`, `NA`, `NaN`, `''`.
  - **Implicit:** values not present in the data.
- **How to deal with missing values?**
  - **Remove** observations or variables with missing values ([`pandas.DataFrame.dropna()`]()).
  - **Replace** missing values (e.g., with 0, with the previous value, with the next value) ([`pandas.DataFrame.fillna()`]()).
  - **Estimate** missing values (e.g., by computing the mean or median value) ([`pandas.Series.interpolate()`]()).
  - **Ignore** missing values.
- Real datasets can also have **inconsistent** or **duplicated values** ([`pandas.DataFrame.drop_duplicates()`]()).

Data Preparation
Exercise 5.2

THINK
PAIR
SHARE

How would you handle missing values in each of the following cases?

1. **A variable has values missing for 45 observations out of 71 observations.**

2. **An observation has a missing value for 1 variable out of 7 variables.**

3. **An observation is missing values for 5 variables out of 7 variables.**

Data Preparation
Aggregation

- **Aggregation** is the combination of two or more observations into a single observation.
  - ***Example*:**
    - Consider a dataset of all transactions in various store locations over the course of a year.
    - We can **aggregate** all the transactions that occur in one day at a specific store into a single daily storewide transaction.
  - **How?**
    - Numeric values can be aggregated by taking a **sum or mean**.
    - Categorical values may be summarized in terms of a higher level category.
  - **Why?**
    - A smaller dataset requires **less memory and processing time** and may enable the use of more expensive algorithms.
    - Provides a **high-level view of the data** instead of a low-level view.
    - **Aggregate** quantities, such as sums or means, have **less variability** than the individual values being **aggregated**.

Data Preparation
Sampling

- **Sampling** is the selection of a subset of the dataset for analysis.
  - The **sample** must be **representative**; that is, it should have approximately the same properties as the original dataset.
  - **Why?**
    - A smaller dataset requires **less memory and processing time** and may enable the use of more expensive algorithms.
  - **How?**
    - **Simple random sampling:** there is an **equal probability** of selecting any particular observation.
      - **Sampling without replacement:** observations are **removed** from the dataset as they are selected for the sample.
      - **Sampling with replacement:** observations are **not removed** from the dataset as they are selected for the sample.
        - The same observation can be selected **more than once**.
    - **Stratified sampling:** observations are selected from each **subgroup** (or **strata**) of the data.
    - **Progressive sampling:** start with a small sample and then increase the sample size until a sample of "**sufficient size**" has been obtained.

Data Preparation
Exercise 5.3

THINK
PAIR
SHARE

For each of the following cases, indicate whether the sampling method is reasonable and why.

1. **Using a random number generator to select 10% of the data.**

2. **Selecting every 10th observation, starting with the first.**

3. **In a transaction log where observations are ordered by time, choosing 10 observations randomly from every 100 observations.**

Data Preparation
Variable Transformation

- A **variable transformation** is a transformation (e.g., **scaling**) that is applied to all the values of a variable.
  - **Why?**
    - Data often includes variables with **different scales** (e.g., age and income). Variables with larger magnitudes may dominate the results of the analysis.
  - **How?**
    - **Standardization** creates a new variable that has a **mean of 0** and a **standard deviation of 1**.

$$x' = \frac{x - \bar{x}}{s_x}$$

      - If data has **outliers**, we often use the **median** and the **absolute standard deviation** instead of the mean and the standard deviation.
    - **Normalization** creates a new variable with range $[0,1]$.

$$x' = \frac{x - min_x}{max_x - min_x}$$

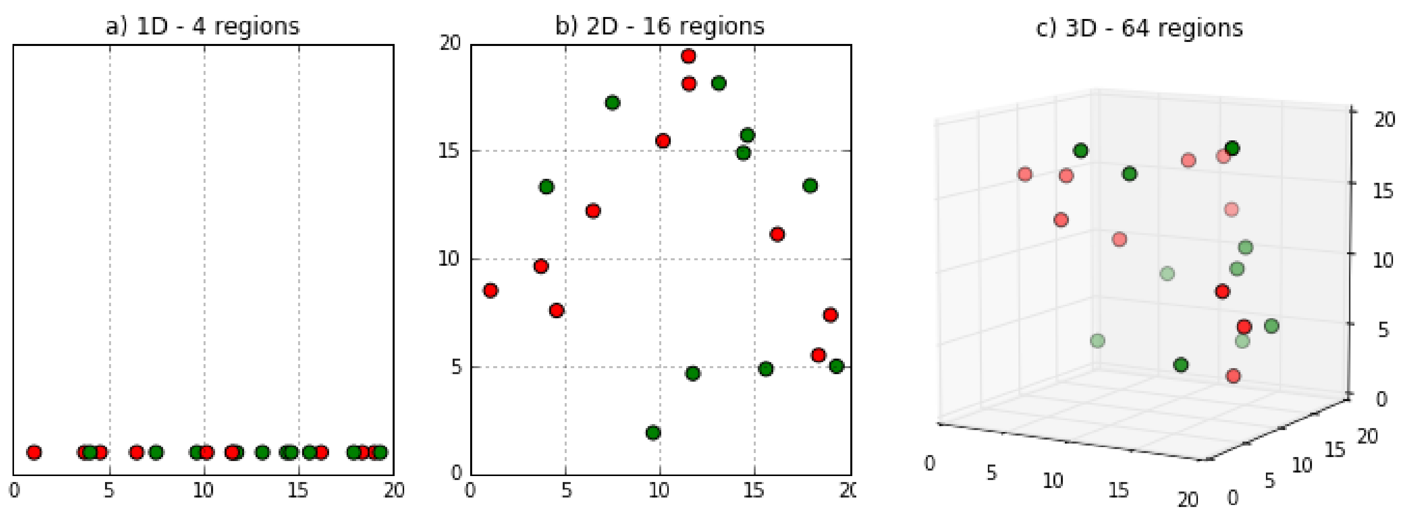
    - **Other transformations:** x^k, $\log x$, e^x, $\sqrt{x}$, $|x|$, etc.

Data Preparation
Dimensionality Reduction

- Datasets can have a large number of variables.
- **Dimensionality reduction** is the **reduction of the number of variables** in the dataset for analysis.
  - **Feature creation** or **feature extraction:** reduce the dimensionality by **creating new variables** that are a combination of the original variables.
  - **Feature selection:** reduce the dimensionality by **selecting a subset of the original variables**.
  - **Why?**
    - Some variables may be **redundant** or **irrelevant**.
    - A model with fewer attributes is usually **easier to understand**.
    - In general, many algorithms work better if the **dimensionality** is lower.
      - **The curse of dimensionality**.

Dimensionality Reduction
The Curse of Dimensionality

- Data analysis becomes significantly harder as the **dimensionality** of the data **increases**.
- As the **dimensionality increases**, the data becomes **increasingly sparse** in the space that it occupies. Thus, the observations in the dataset are **not a representative sample** of all possible observations.



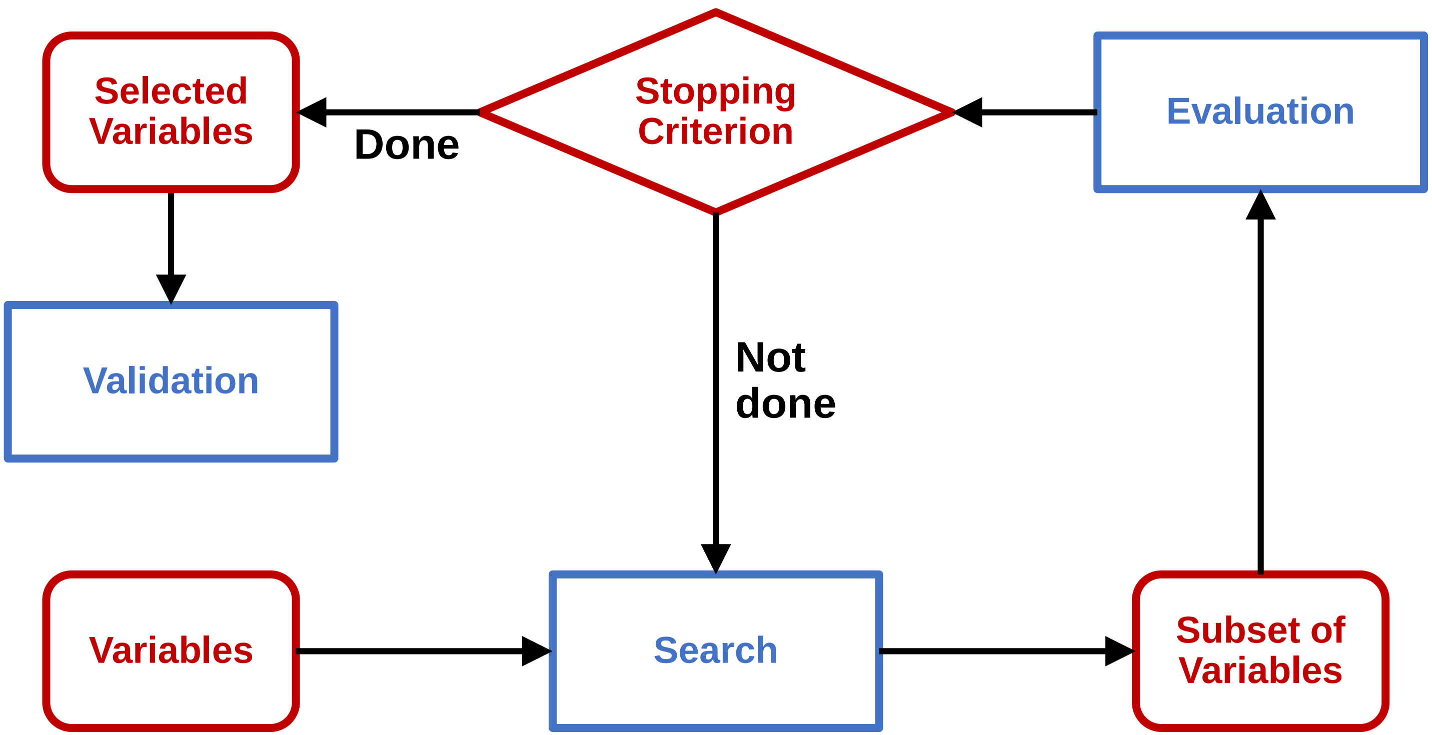
Source: KDnuggets, *Must-Know: What is the curse of dimensionality?* (2017)

Data Preparation
Feature Selection (I)

- **Feature selection** is the reduction of the number of variables by **selecting a subset of the original variables**.
  - **Univariate feature selection:** select the "best" k variables.
    - In Python (with **sklearn**): `sklearn.feature_selection.SelectKBest(…)`
    - **What are the disadvantages of this approach?**
  - **Multivariate feature selection:** select the "best" **subset** of variables.
    - The ideal approach is to try all possible subsets of variables and select the one that produces the best results.
      - **Is this feasible?**
    - Other approaches:
      - **Embedded approaches**: feature selection is part of the data analysis algorithms (e.g., **decision trees**).
      - **Filter approaches:** variables are selected **before running the data analysis algorithm** using some approach that is independent of the data analysis task.
      - **Wrapper approaches:** the results of the data analysis algorithm is used to select the "best" subset of variables.

Data Preparation
Feature Selection (II)

- **Feature selection** is the reduction of the number of variables by **selecting a subset of the original variables**.
  - **Multivariate feature selection:** select the “best” **subset** of variables.



Data Preparation
References

- Daniel Chen. *Pandas for Everyone* (2018).
- Garrett Grolemund and Hadley Wickham. *R for Data Science* (2016).
- Joel Grus. *Data Science from Scratch* (2015).
- Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. *Introduction to Data Mining* (2019).
- Hadley Wickham. *Tidy Data* (2014).