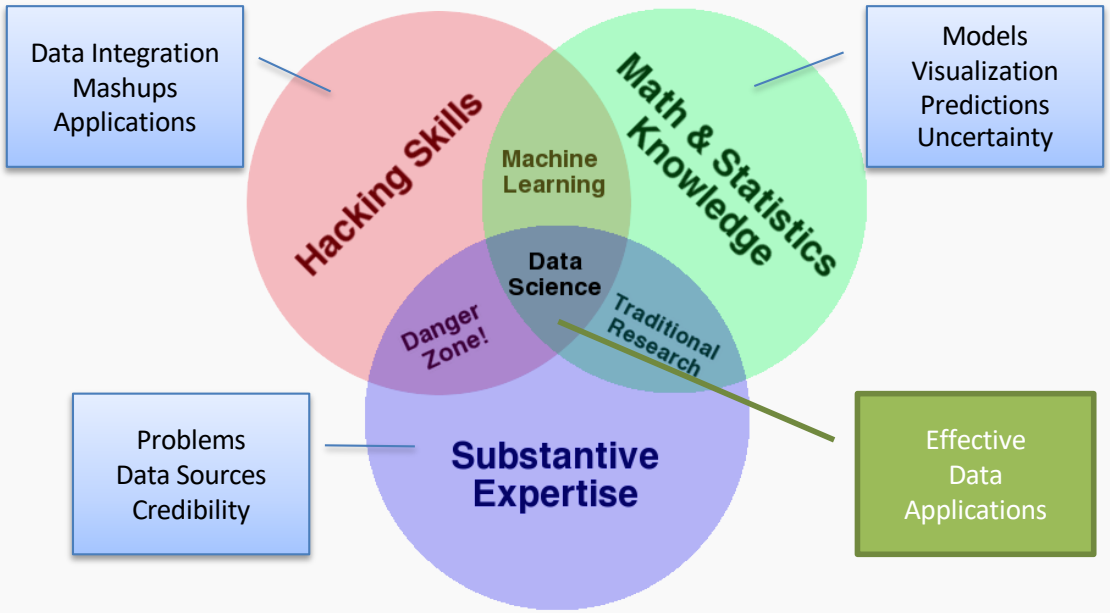
Data Science Process

Three Essential Skills of Data Scientists and Advanced Analytics



Why “Danger Zone?”

- Ronny Kohavi* keynote at KDD 2015
- People are incredibly clever at explaining “very surprising results”. Unfortunately most very surprising results are caused by data pipeline errors.

- Beware “HiPPOs” (Highest Paid-Person’s Opinion)
  – Listen to the customers and don’t let the HiPPO kill good ideas

* General Manager for Microsoft’s Analysis and Experimentation Team

Personalized Correlated Recommendations

- Actual personalized recommendations from Amazon.

- Buy anti aging serum because you bought an LED light bulb (Maybe the wrinkles show?)
- Buy Atonement movie DVD because you bought a Maglite flashlight (must be a dark movie)
- Buy Organic Virgin Olive Oil because you bought Toilet Paper. (If there is causality here, it's probably in the other direction.)

Main Categories of Data

In data science and big data, you'll come across many different types of data, and each of them tends to require different tools and techniques. The main categories of data are these:

- Structured
- Unstructured
- Natural language
- Machine-generated
- Graph-based
- Audio, video, and images
- Streaming

Structured data

	Indicator ID	Dimension List	Timeframe	Numeric Value	Missing Value Flag	Confidence Inte
2	214390830	Total (Age-adjusted)	2008	74.6%		73.8%
3	214390833	Aged 18-44 years	2008	59.4%		58.0%
4	214390831	Aged 18-24 years	2008	37.4%		34.6%
5	214390832	Aged 25-44 years	2008	66.9%		65.5%
6	214390836	Aged 45-64 years	2008	88.6%		87.7%
7	214390834	Aged 45-54 years	2008	86.3%		85.1%
8	214390835	Aged 55-64 years	2008	91.5%		90.4%
9	214390840	Aged 65 years and over	2008	94.6%		93.8%
10	214390837	Aged 65-74 years	2008	93.6%		92.4%
11	214390838	Aged 75-84 years	2008	95.6%		94.4%
12	214390839	Aged 85 years and over	2008	96.0%		94.0%
13	214390841	Male (Age-adjusted)	2008	72.2%		71.1%
14	214390842	Female (Age-adjusted)	2008	76.8%		75.9%
15	214390843	White only (Age-adjusted)	2008	73.8%		72.9%
16	214390844	Black or African American only (Age-adjusted)	2008	77.0%		75.0%
17	214390845	American Indian or Alaska Native only (Age-adjusted)	2008	66.5%		57.1%
18	214390846	Asian only (Age-adjusted)	2008	80.5%		77.7%
19	214390847	Native Hawaiian or Other Pacific Islander only (Age-adjusted)	2008	DSU		
20	214390848	2 or more races (Age-adjusted)	2008	75.6%		69.6%

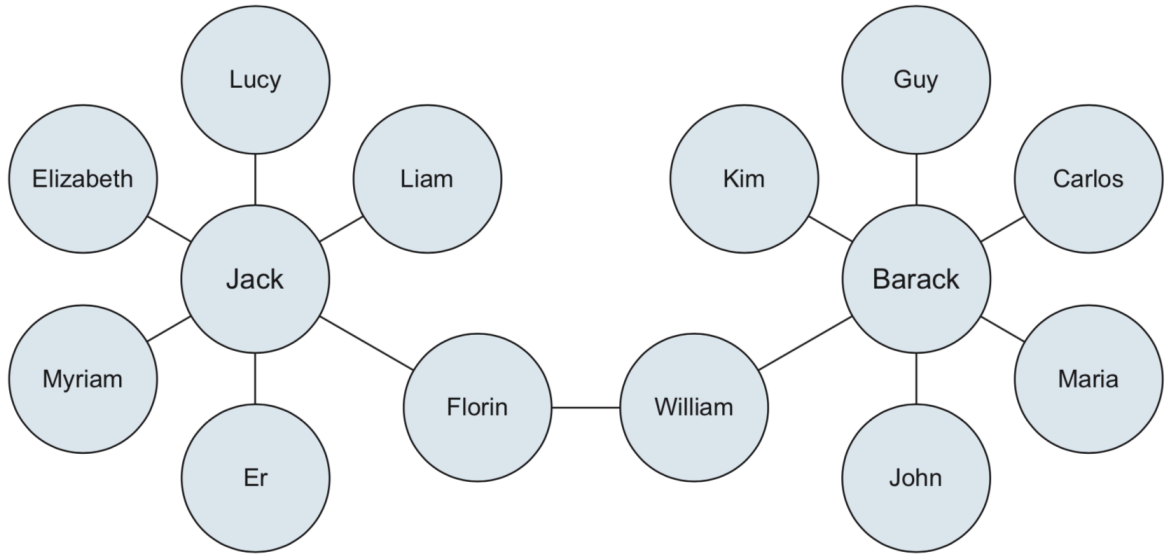
Machine-generated data

Machine-generated data is information that's automatically created by a computer, process, application, or other machine without human intervention. Examples of machine data are web server logs, call detail records, network event logs, and telemetry.

```
CSIPERF:TXCOMMIT;313236
2014-11-28 11:36:13, Info                  CSI    00000153 Creating NT transaction (seq
69), objectname [6]"(null)"
2014-11-28 11:36:13, Info                  CSI    00000154 Created NT transaction (seq 69)
result 0x00000000, handle @0x4e54
2014-11-28 11:36:13, Info                  CSI    00000155@2014/11/28:10:36:13.471
Beginning NT transaction commit...
2014-11-28 11:36:13, Info                  CSI    00000156@2014/11/28:10:36:13.705 CSI perf
trace:
CSIPERF:TXCOMMIT;273983
2014-11-28 11:36:13, Info                  CSI    00000157 Creating NT transaction (seq
70), objectname [6]"(null)"
2014-11-28 11:36:13, Info                  CSI    00000158 Created NT transaction (seq 70)
result 0x00000000, handle @0x4e5c
2014-11-28 11:36:13, Info                  CSI    00000159@2014/11/28:10:36:13.764
Beginning NT transaction commit...
2014-11-28 11:36:14, Info                  CSI    0000015a@2014/11/28:10:36:14.094 CSI perf
trace:
CSIPERF:TXCOMMIT;386259
2014-11-28 11:36:14, Info                  CSI    0000015b Creating NT transaction (seq
71), objectname [6]"(null)"
2014-11-28 11:36:14, Info                  CSI    0000015c Created NT transaction (seq 71)
result 0x00000000, handle @0x4e5c
2014-11-28 11:36:14, Info                  CSI    0000015d@2014/11/28:10:36:14.106
Beginning NT transaction commit...
2014-11-28 11:36:14, Info                  CSI    0000015e@2014/11/28:10:36:14.428 CSI perf
trace:
CSIPERF:TXCOMMIT;375581
```

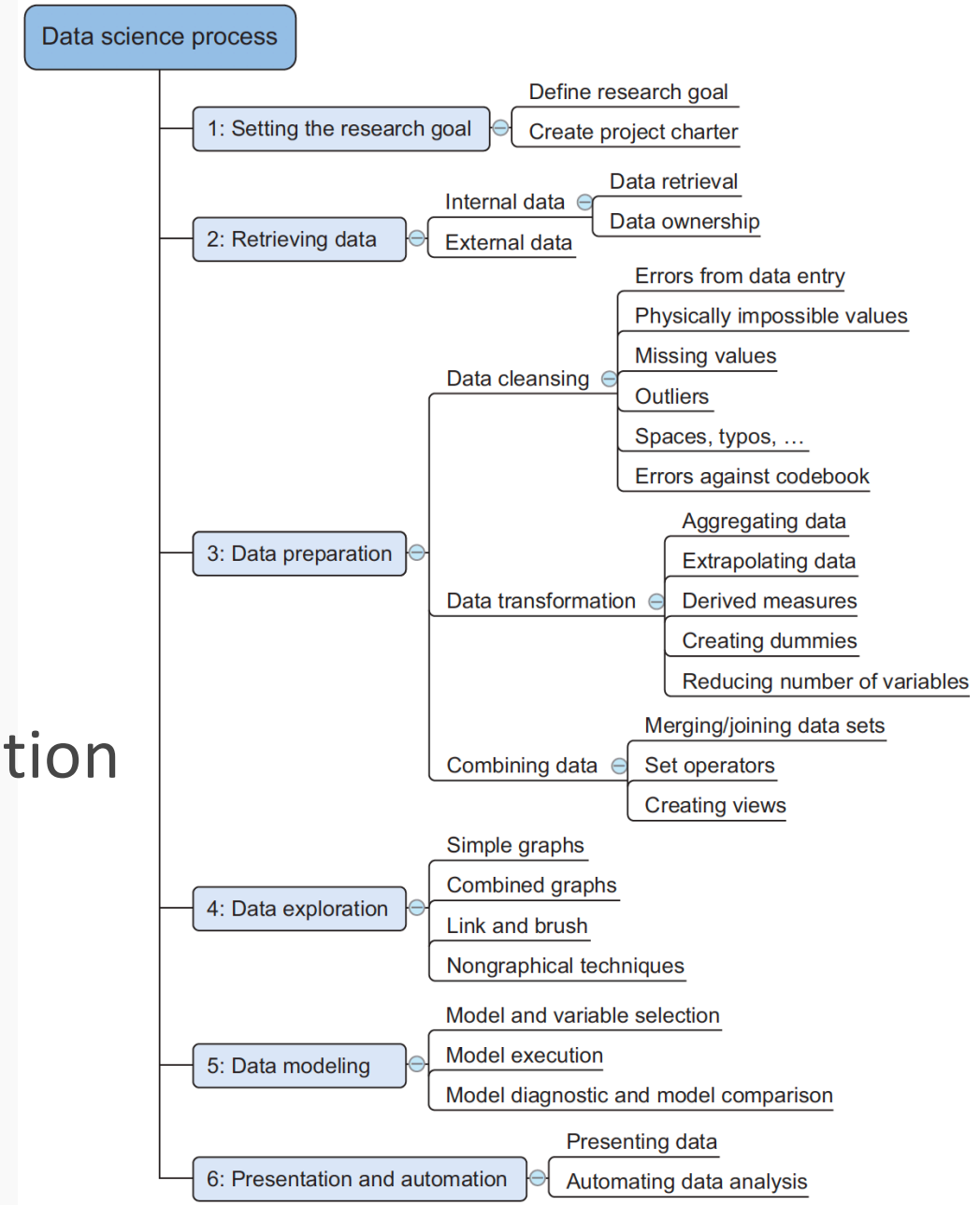
Graph-based or network data

- "Graph" in this case points to mathematical *graph theory*. In graph theory, a graph is a mathematical structure to model pair-wise relationships between objects.
- Graph or network data is, in short, data that focuses on the relationship or adjacency of objects.
- The graph structures use nodes, edges, and properties to represent and store graph data.
- Graph-based data is a natural way to represent social networks, and its structure allows you to calculate specific metrics such as the influence of a person and the shortest path between two people.
- Graph databases are used to store graph-based data and are queried with specialized query languages such as SPARQL.



The data science process

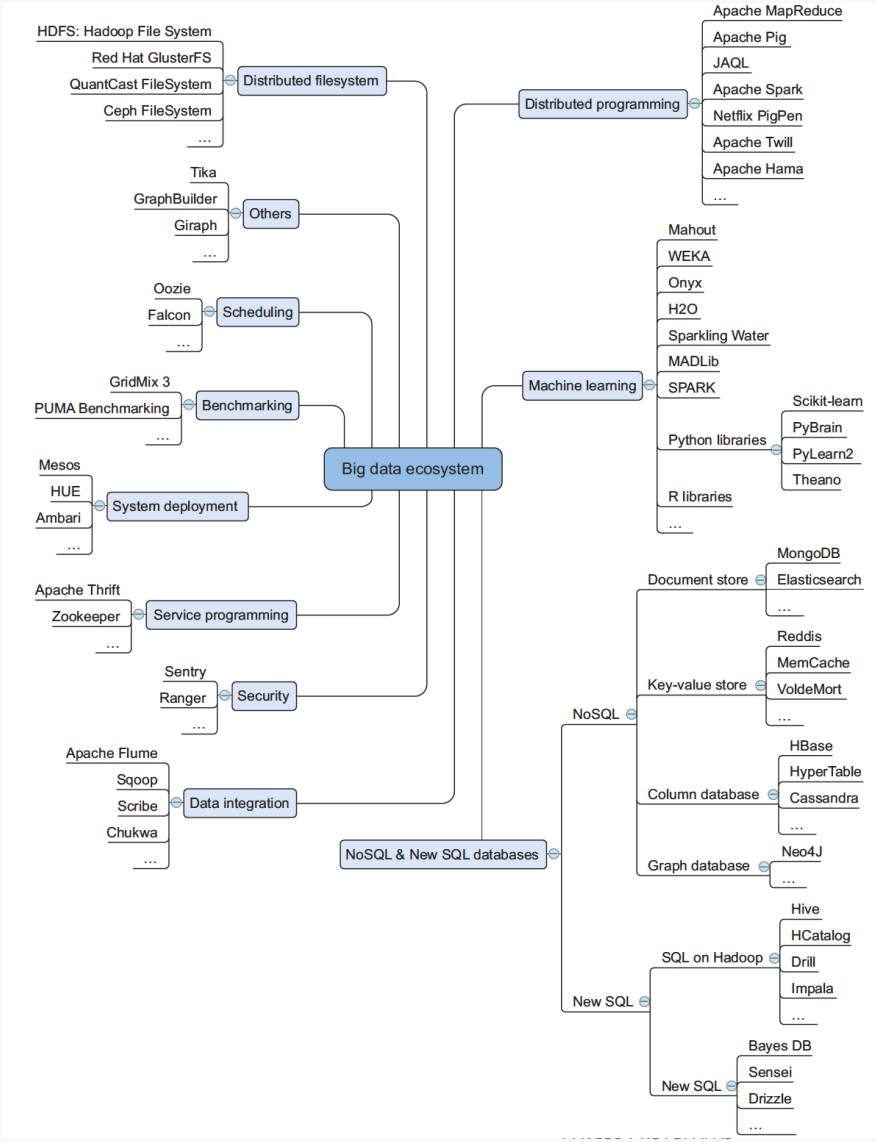
- Setting the research goal
- Retrieving data
- Data preparation
- Data exploration
- Data modeling
- Presentation and automation



- **This is AN ITERATIVE PROCESS!**

The big data ecosystem and Data Science

- The big data landscape is more than Hadoop alone. It consists of many different technologies
  - File system
  - Distributed programming frameworks
  - Data integration
  - Databases
  - Machine learning
  - Security
  - Scheduling
  - Benchmarking
  - System deployment
  - Service programming
- Not every big data category is utilized heavily by data scientists.
  - They focus mainly on the file system, the distributed programming frameworks, databases



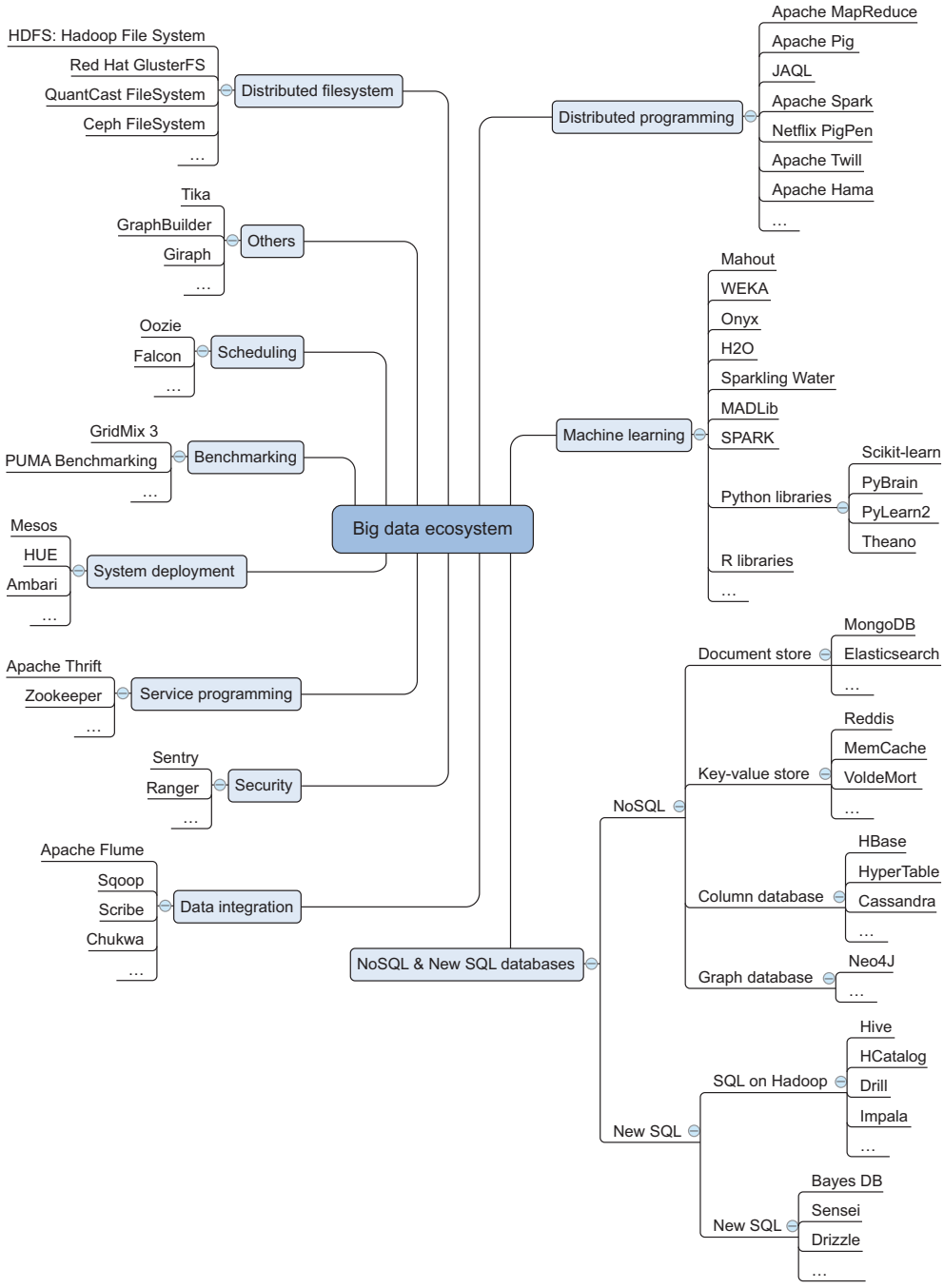
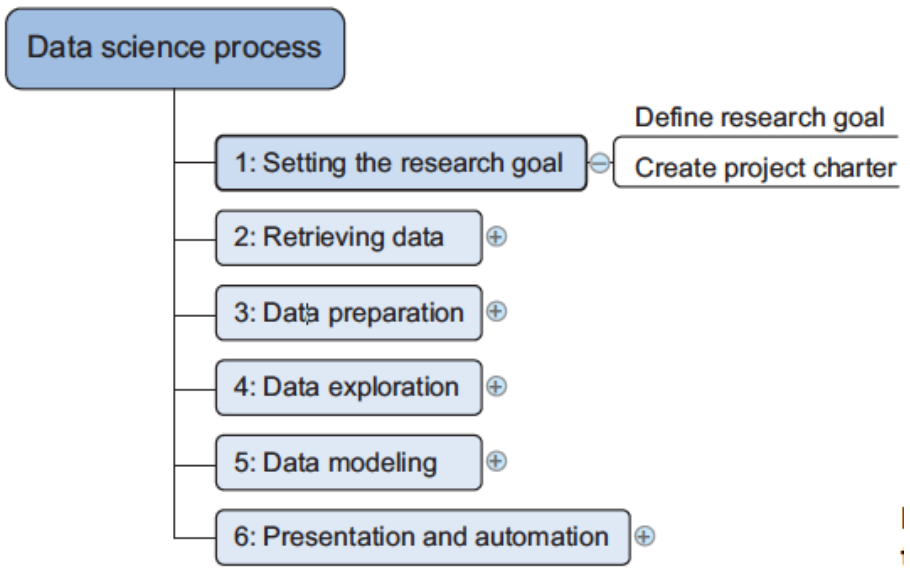


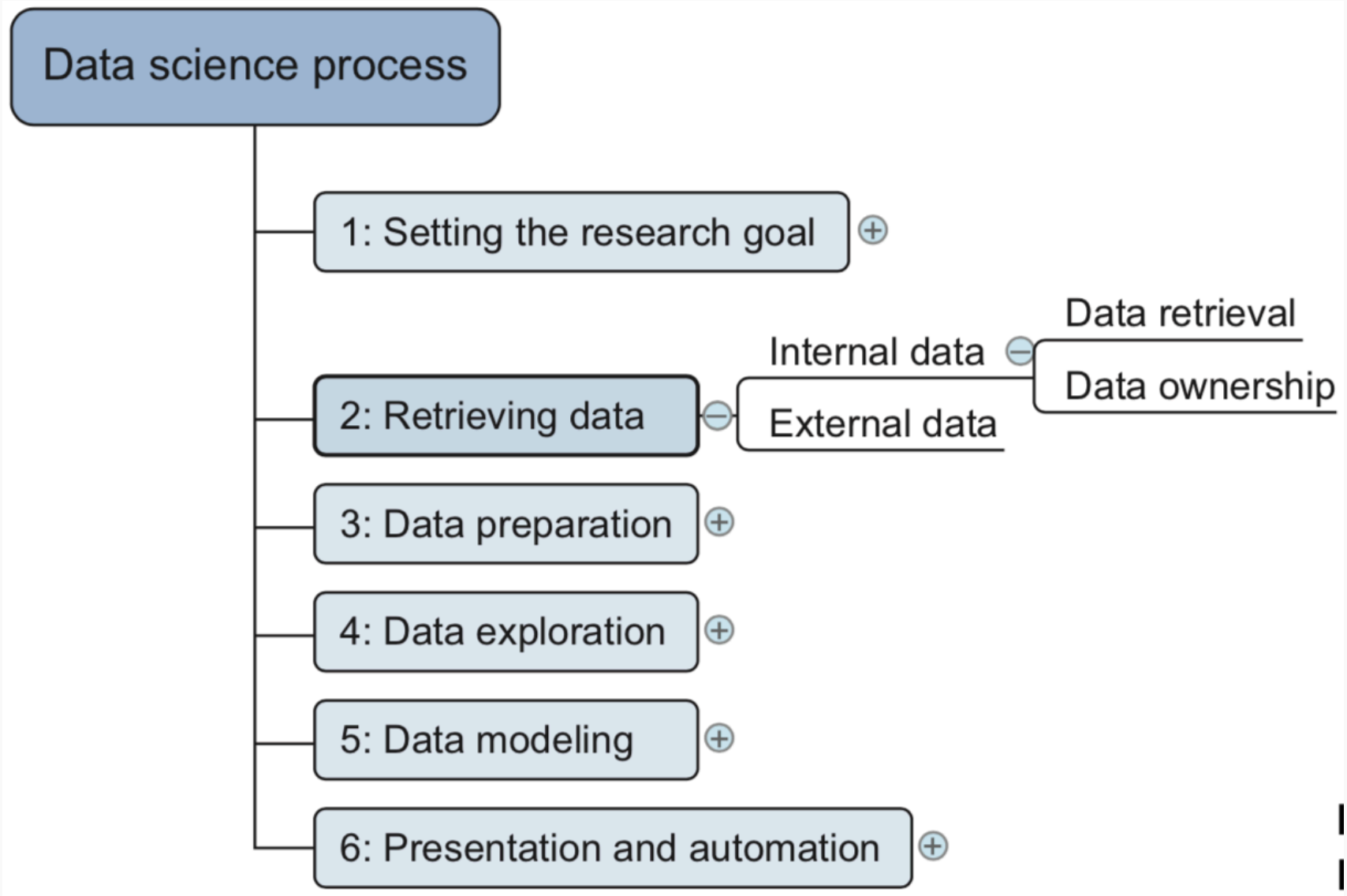
Figure 1.6 Big data technologies can be classified into a few main components.

Step 1: Defining research goals and creating a project charter

- Spend time understanding the goals and context of your research
  - Many data scientists fail here: despite their mathematical wit and scientific brilliance, they never seem to grasp the business goals and context
- Create a project charter
  - A clear research goal
  - The project mission and context
  - How you're going to perform your analysis
  - What resources you expect to use
  - Proof that it's an achievable project, or proof of concepts
  - Deliverables and a measure of success
  - A timeline



Step 2: Retrieving data



Internal Data

- Assess the relevance and quality of the data that's readily available within your company.
- Most companies have a program for maintaining key data, so much of the cleaning work may already be done.
- This data can be stored in official data repositories such as **databases, data marts, data warehouses,** and **data lakes**
- The possibility exists that your data still resides in Excel or text files on the desktop of a domain expert.

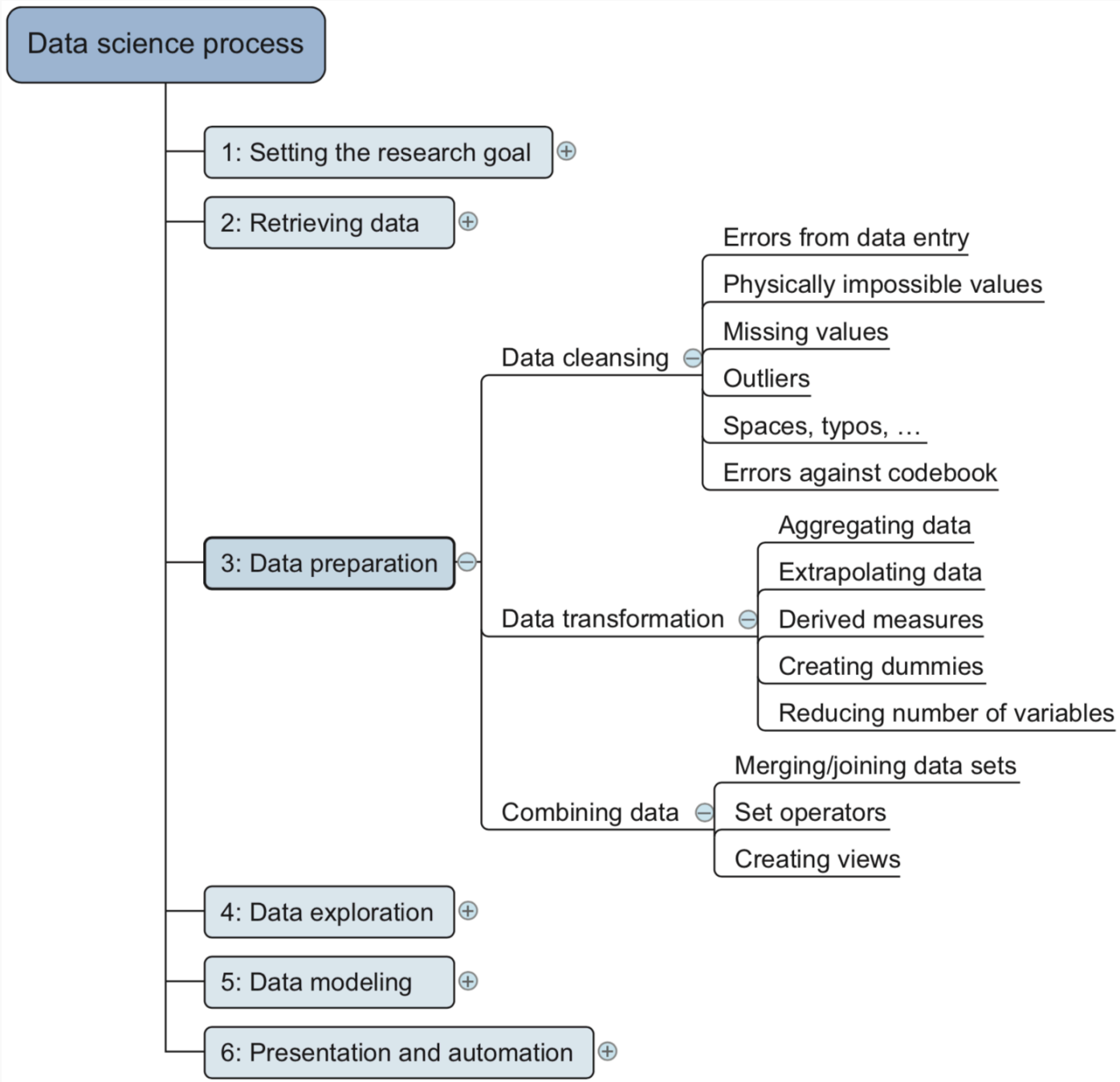
- Getting access to data is another difficult task. Organizations understand the value and sensitivity of data and often have policies in-place, so everyone has access to what they need and nothing more.

External Data

- If data isn't available inside your organization, look outside your organization.
- Many companies specialize in collecting valuable information. For instance, Nielsen and GFK are well known for this in the retail industry. Other companies provide data so that you, in turn, can enrich their services and ecosystem. Such is the case with Twitter, LinkedIn, and Facebook.
- Open Data

Open data site	Description
Data.gov	The home of the US Government's open data
https://open-data.europa.eu/	The home of the European Commission's open data
Freebase.org	An open database that retrieves its information from sites like Wikipedia, MusicBrains, and the SEC archive
Data.worldbank.org	Open data initiative from the World Bank
Aiddata.org	Open data for international development
Open.fda.gov	Open data from the US Food and Drug Administration

Step 3: Cleansing, integrating, and transforming data



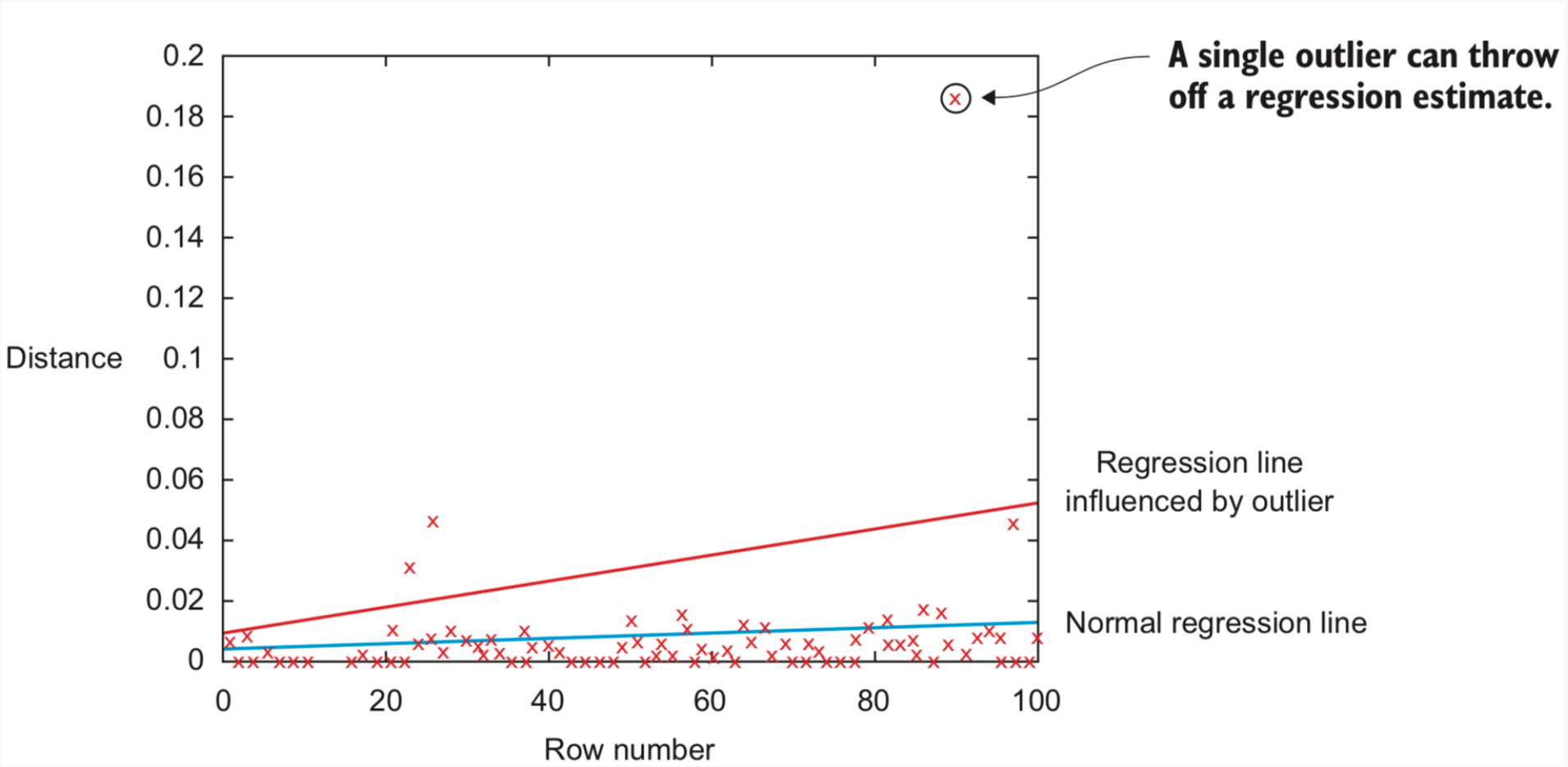
Cleansing data

- Data cleansing is a subprocess of the data science process that focuses on removing errors in your data, so that your data becomes a true and consistent representation of the processes it originates from.
- 2 types of errors
  - *interpretation error*, such as saying that a person's age is greater than 300 years
  - *inconsistencies* (examples: putting "Female" in one table and "F" in another or using Pounds in one table and Dollars in another.)

An overview of common errors

General solution	
Try to fix the problem early in the data acquisition chain or else fix it in the program.	
Error description	**Possible solution**
Errors pointing to false values within one data set	
Mistakes during data entry	Manual overrules
Redundant white space	Use string functions
Impossible values	Manual overrules
Missing values	Remove observation or value
Outliers	Validate and, if erroneous, treat as missing value (remove or insert)
Errors pointing to inconsistencies between data sets	
Deviations from a code book	Match on keys or else use manual overrules
Different units of measurement	Recalculate
Different levels of aggregation	Bring to same level of measurement by aggregation or extrapolation

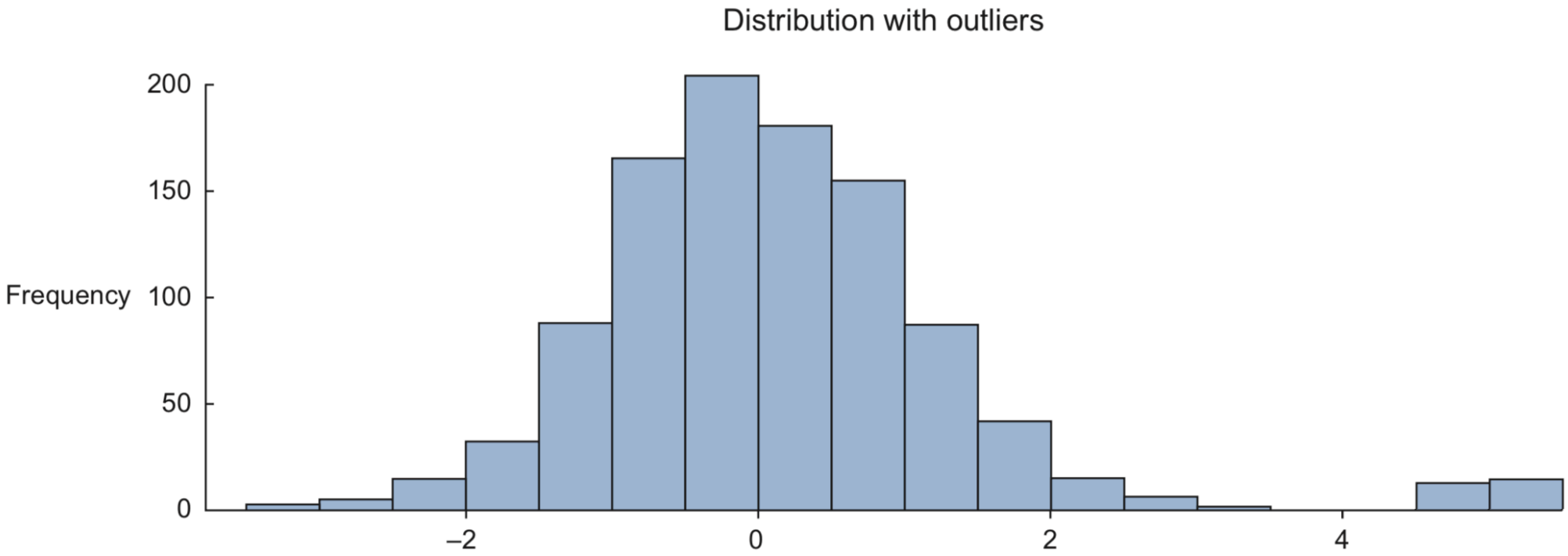
Outliers



The encircled point influences the model heavily and is worth investigating because it can point to a region where you don't have enough data or might indicate an error in the data, but it also can be a valid data point.

Outliers

- An outlier is an observation that seems to be distant from other observations or, more specifically, one observation that follows a different logic or generative process than the other observations.
- Outliers can critically influence your data modeling, so investigate them first.
- The easiest way to find outliers is to use a plot or a histogram with the distribution of frequencies of occurrences between minimum and maximum values.



DATA ENTRY ERRORS

- Data collection and data entry are error-prone processes. They often require human intervention, and because humans are only human, they make typos or lose their concentration for a second and introduce an error into the chain.
- But data collected by machines or computers isn't free from errors either
  - due to machine or hardware failure
  - transmission errors or bugs in the extract, transform, and load phase (ETL)

Detecting outliers on simple variables with a frequency table

Value	Count
Good	1598647
Bad	1354468
Godo	15
Bade	1

Most errors of this type are easy to fix with simple assignment statements and if-then-else rules:

```
if x == “Godo”:
    x = “Good”
if x == “Bade”:
    x = “Bad”
```

REDUNDANT WHITESPACE

- Whitespaces tend to be hard to detect but cause errors like other redundant characters would.
- Example: The cleaning during the ETL phase wasn’t well executed, and keys in one table contained a whitespace at the end of a string. This caused a mismatch of keys such as “FR ” – “FR”, dropping the observations that couldn’t be matched.
- Fixing redundant whitespaces is luckily easy enough in most programming languages. They all provide string functions that will remove the leading and trailing whitespaces.
- For instance, in Python you can use the strip() function to remove leading and trailing spaces.

FIXING CAPITAL LETTER MISMATCHES

- Capital letter mismatches are common. Most programming languages make a distinction between “Brazil” and “brazil”.
- In this case you can solve the problem by applying a function that returns both strings in lowercase, such as .lower() in Python.
- “Brazil”.lower() == “brazil”.lower() should result in true.

IMPOSSIBLE VALUES AND SANITY CHECKS

- Sanity checks are another valuable type of data check.
- Here you check the value against physically or theoretically impossible values such as:
  - people taller than 3 meters or
  - someone with an age of 299 years.

- Sanity checks can be directly expressed with rules:

check = 0 <= age <= 120

DEALING WITH MISSING VALUES

- Missing values aren’t necessarily wrong, but you still need to handle them separately.
- Certain modeling techniques can’t handle missing values.
- They might be an indicator that something went wrong in your data collection or that an error happened in the ETL process.

An overview of techniques to handle missing data

Technique	Advantage	Disadvantage
Omit the values	Easy to perform	You lose the information from an observation
Set value to `null`	Easy to perform	Not every modeling technique and/or implementation can handle `null` values
Impute a static value such as 0 or the mean	Easy to perform You don't lose information from the other variables in the observation	Can lead to false estimations from a model
Impute a value from an estimated or theoretical distribution	Does not disturb the model as much	Harder to execute You make data assumptions
Modeling the value (nondependent)	Does not disturb the model too much	Can lead to too much confidence in the model Can artificially raise dependence among the variables Harder to execute You make data assumptions

DIFFERENT UNITS OF MEASUREMENT OR DIFFERENT FORMAT

- When integrating two data sets, you have to pay attention to their respective units of measurement.
- An example of this would be when you study the prices of gasoline in the world. To do this you gather data from different data providers. Data sets can contain prices per gallon and others can contain prices per liter. A simple conversion will do the trick in this case.
- Many errors are due to different formats of the dates or numbers
- Examples: 12/10/2021 can mean 12. October 2021 in most of Europe or 10. December 2021 in US and UK format

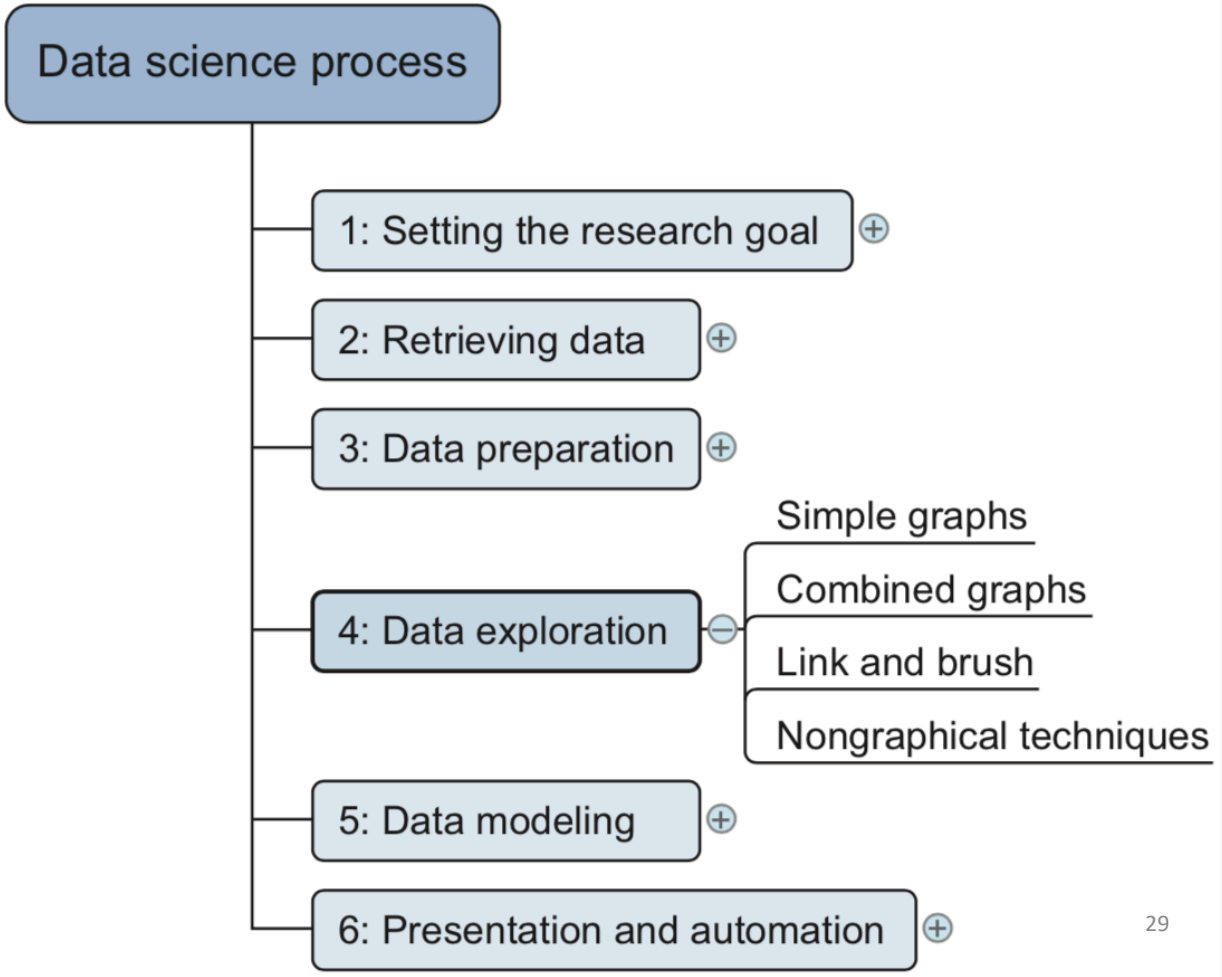
DIFFERENT LEVELS OF AGGREGATION

- Having different levels of aggregation is similar to having different types of measurement.
- An example of this would be a data set containing data per week versus one containing data per work week.
- This type of error is generally easy to detect, and *summarizing* (or the inverse, *expanding*) the data sets will fix it.

Step 4: Exploratory data analysis

During exploratory data analysis you take a deep dive into the data (see figure 2.14). Information becomes much easier to grasp when shown in a picture, therefore you mainly use graphical techniques to gain an understanding of your data and the inter- actions between variables.

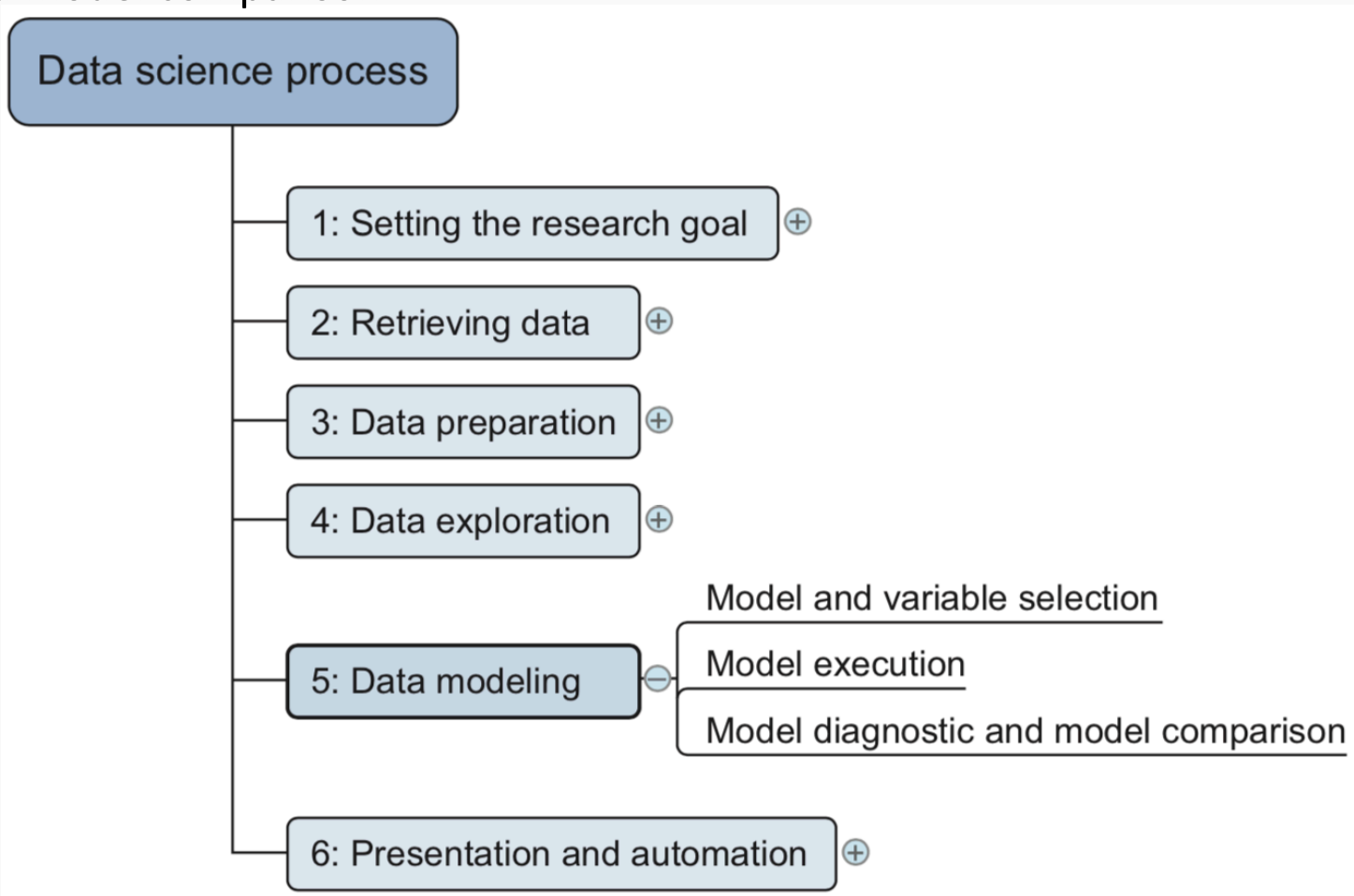
The techniques used in this phase are mainly visual, but in practice they're certainly not limited to visualization techniques. Tabulation, clustering, and other modeling techniques can also be a part of exploratory analysis. Even building simple models can be a part of this step.



Step 5: Build the models

Building a model is an iterative process. The way you build your model depends on whether you go with classic statistics or machine learning, and the type of technique you want to use. Either way, most models consist of the following main steps:

1. Selection of a modeling technique and variables to enter in the model
2. Execution of the model
3. Diagnosis and model comparison



Model and variable selection

- It is needed to select the variables you want to include in your model and a modeling technique.
- The findings from the exploratory analysis should already give a fair idea of what variables will help you construct a good model.
- Many modeling techniques are available and choosing the right model for a problem requires judgment on your part. You'll need to consider model performance and whether your project meets all the requirements to use your model, as well as other factors:
  - Must the model be moved to a production environment and, if so, would it be easy to implement?
  - How difficult is the maintenance on the model: how long will it remain relevant if left untouched?
  - Does the model need to be easy to explain?

Model execution

- Once you've chosen a model you'll need to implement it in code.
- Luckily, most programming languages, such as Python, already have libraries such as StatsModels or Scikit-learn.
- These packages use several of the most popular techniques. Coding a model is a nontrivial task in most cases, so having these libraries available can speed up the process.

Listing 2.1 Executing a linear prediction model on semi-random data

```
import statsmodels.api as sm
import numpy as np
predictors = np.random.random(1000).reshape(500,2)
target = predictors.dot(np.array([0.4, 0.6])) + np.random.random(500)
lmRegModel = sm.OLS(target,predictors)
result = lmRegModel.fit()
result.summary()
```

Imports required Python modules.

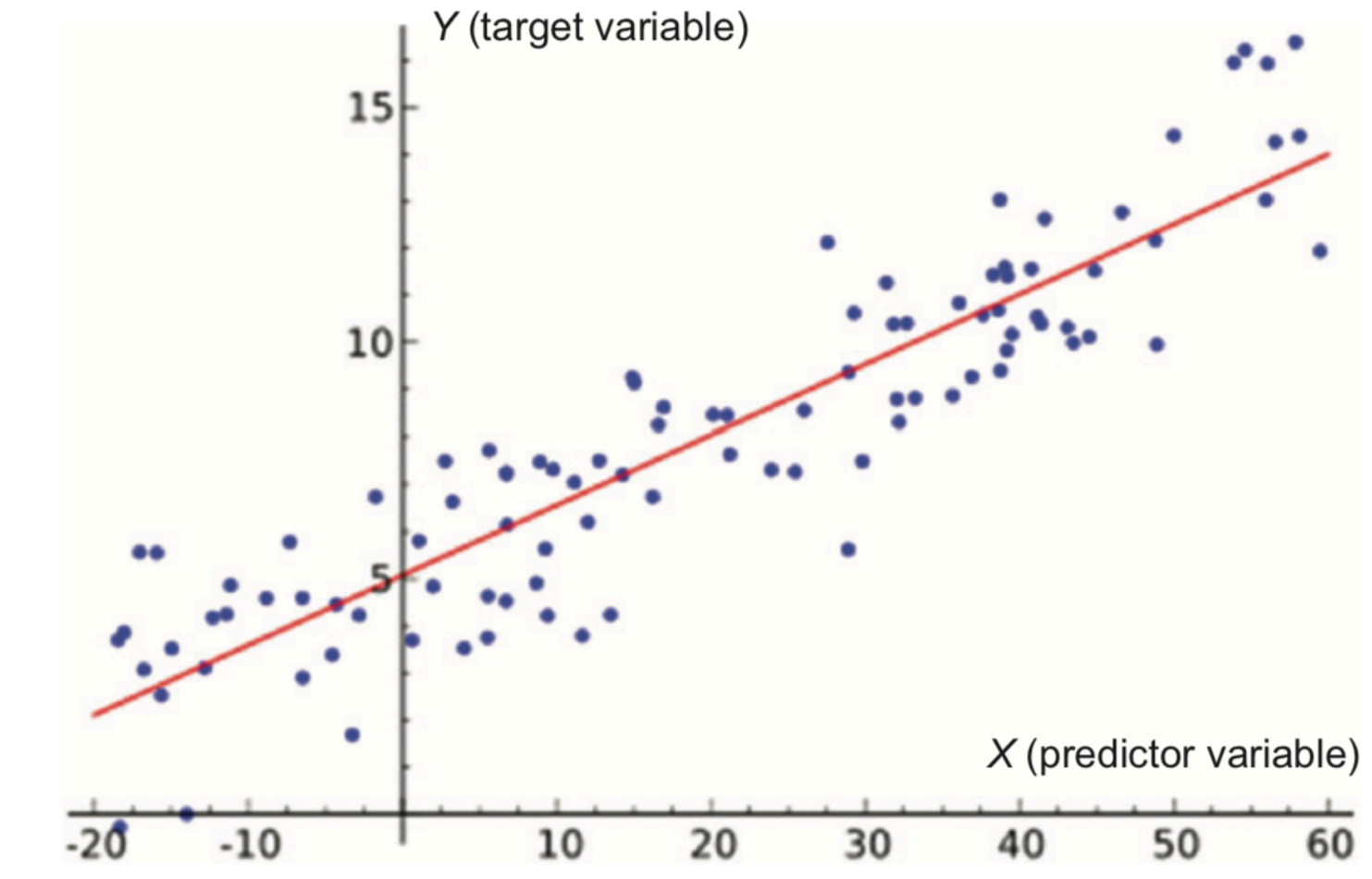
Creates random data for predictors (x-values) and semi-random data for the target (y-values) of the model. We use predictors as input to create the target so we infer a correlation here.

Fits linear regression on data.

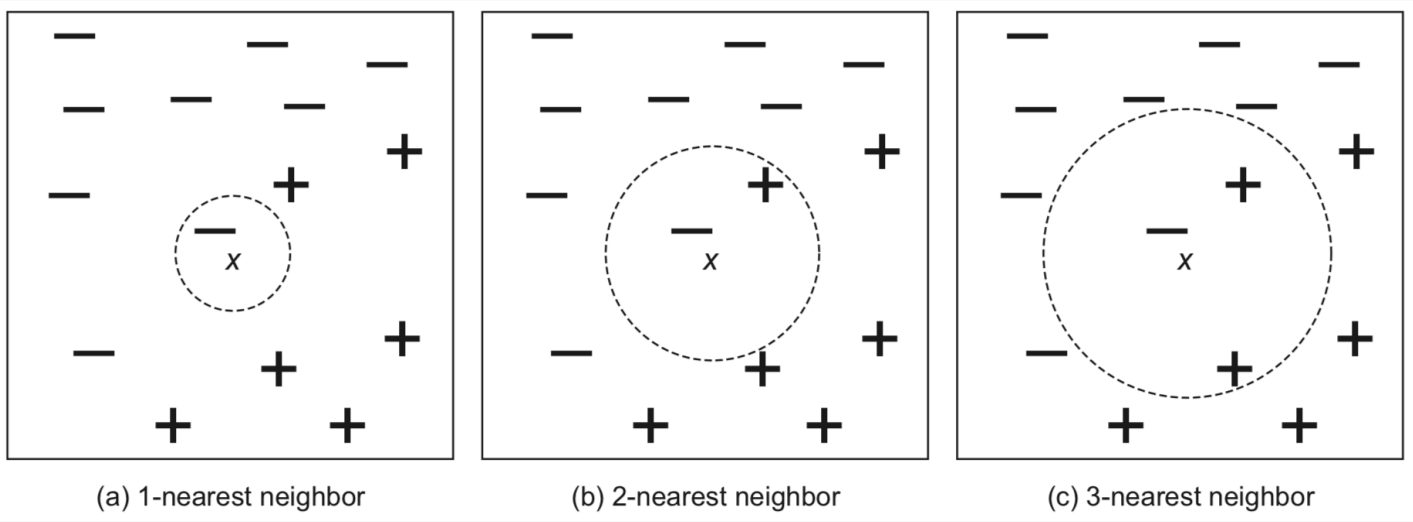
Shows model fit statistics.

Model execution (Linear Prediction example)

Linear regression tries to fit a line while minimizing the distance to each point



Model execution (K-nearest neighbor technique example)



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Listing 2.2 Executing k-nearest neighbor classification on semi-random data

```
from sklearn import neighbors
predictors = np.random.random(1000).reshape(500,2)
target = np.around(predictors.dot(np.array([0.4, 0.6])) +
         np.random.random(500))
clf = neighbors.KNeighborsClassifier(n_neighbors=10)
knn = clf.fit(predictors,target)
knn.score(predictors, target)
```

Imports modules.

Creates random predictor data and semi-random target data based on predictor data.

Fits 10-nearest neighbors model.

Gets model fit score: what percent of the classification was correct?

Model execution (K-nearest neighbor technique example)

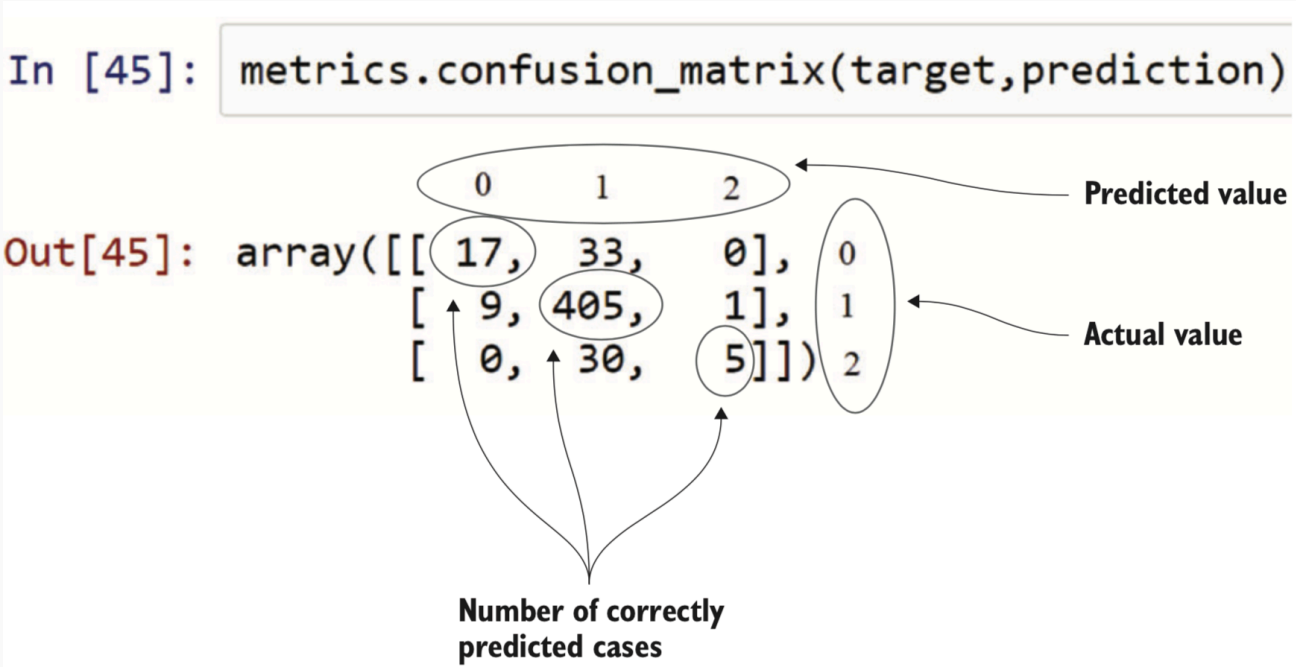
- by “scoring a model” we often mean applying it on data to make a prediction.

prediction = knn.predict(predictors)

- Now we can use the prediction and compare it to the real thing using a confusion matrix.

metrics.confusion_matrix(target,prediction)

- We get a 3-by-3 matrix as



Confusion matrix: it shows how many cases were correctly classified and incorrectly classified by comparing the prediction with the real values. Remark: the classes (0,1,2) were added in the figure for clarification.

Model diagnostics and model comparison

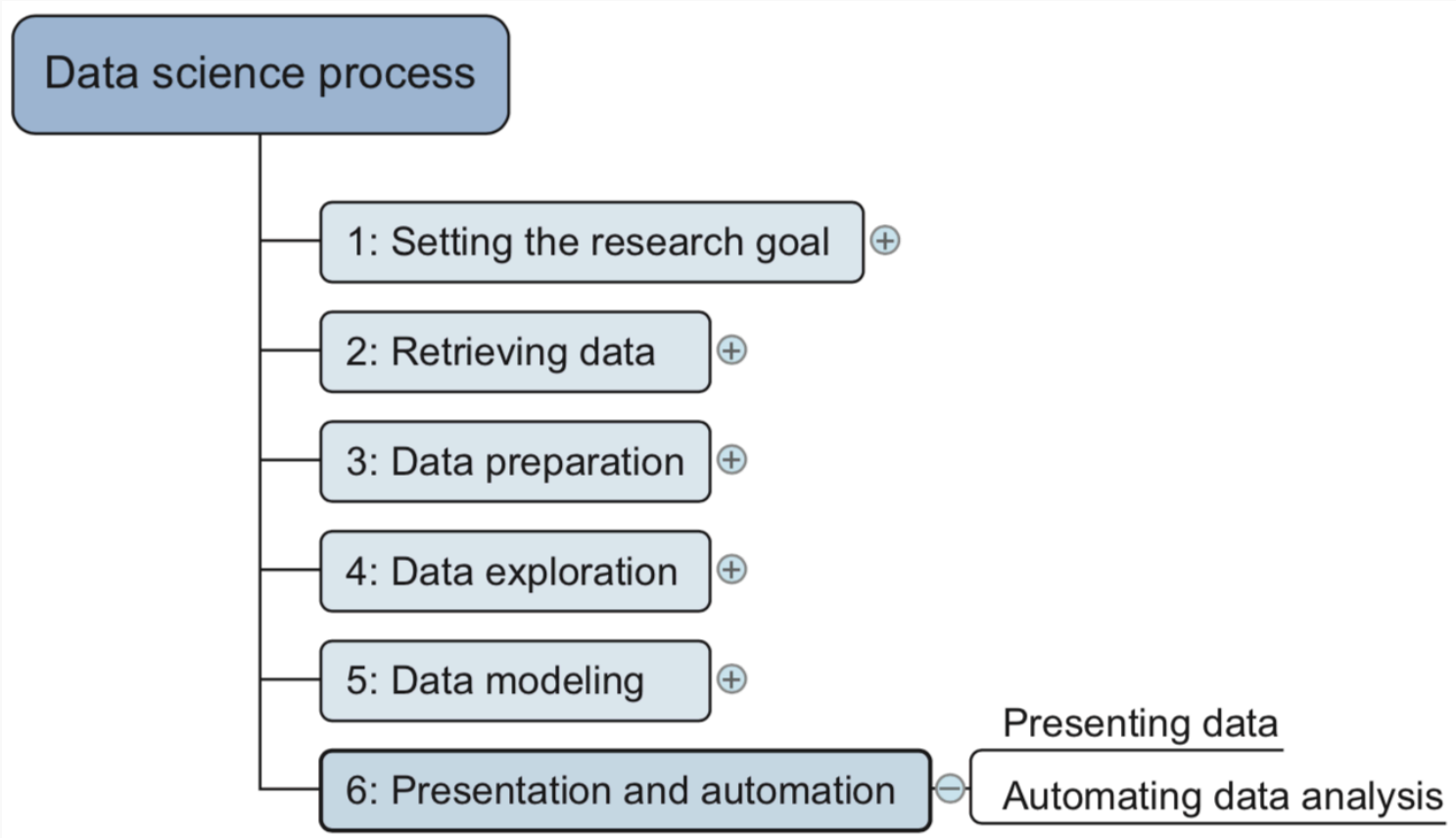
- You'll be building multiple models from which you then choose the best one based on multiple criteria.
- Working with a holdout sample helps you pick the best-performing model. A holdout sample is a part of the data you leave out of the model building so it can be used to evaluate the model afterward.
- The principle here is simple: the model should work on unseen data. You use only a fraction of your data to estimate the model and the other part, the holdout sample, is kept for later use. The model is then unleashed on the unseen data and error measures are calculated to evaluate it.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

Formula for mean square error

Mean square error is a simple measure: check for every prediction how far it was from the truth, square this error, and add up the error of every prediction.

Step 6: Presenting findings and building applications on top of them



Sometimes people get so excited about your work that you'll need to repeat it over and over again because they value the predictions of your models or the insights that you produced. For this reason, you need to automate your models. This doesn't always mean that you have to redo all of your analysis all the time. Sometimes it's sufficient that you implement only the model scoring; other times you might build an application that automatically updates reports, Excel spreadsheets, or PowerPoint presentations.

The data science process

- Setting the research goal
- Retrieving data
- Data preparation
- Data exploration
- Data modeling
- Presentation and automation