The Data Science Process

What is Data Science?

- Data science or Data analytics is the process of analyzing large set of data points to get answers on questions related to that data sets

What is Pandas?

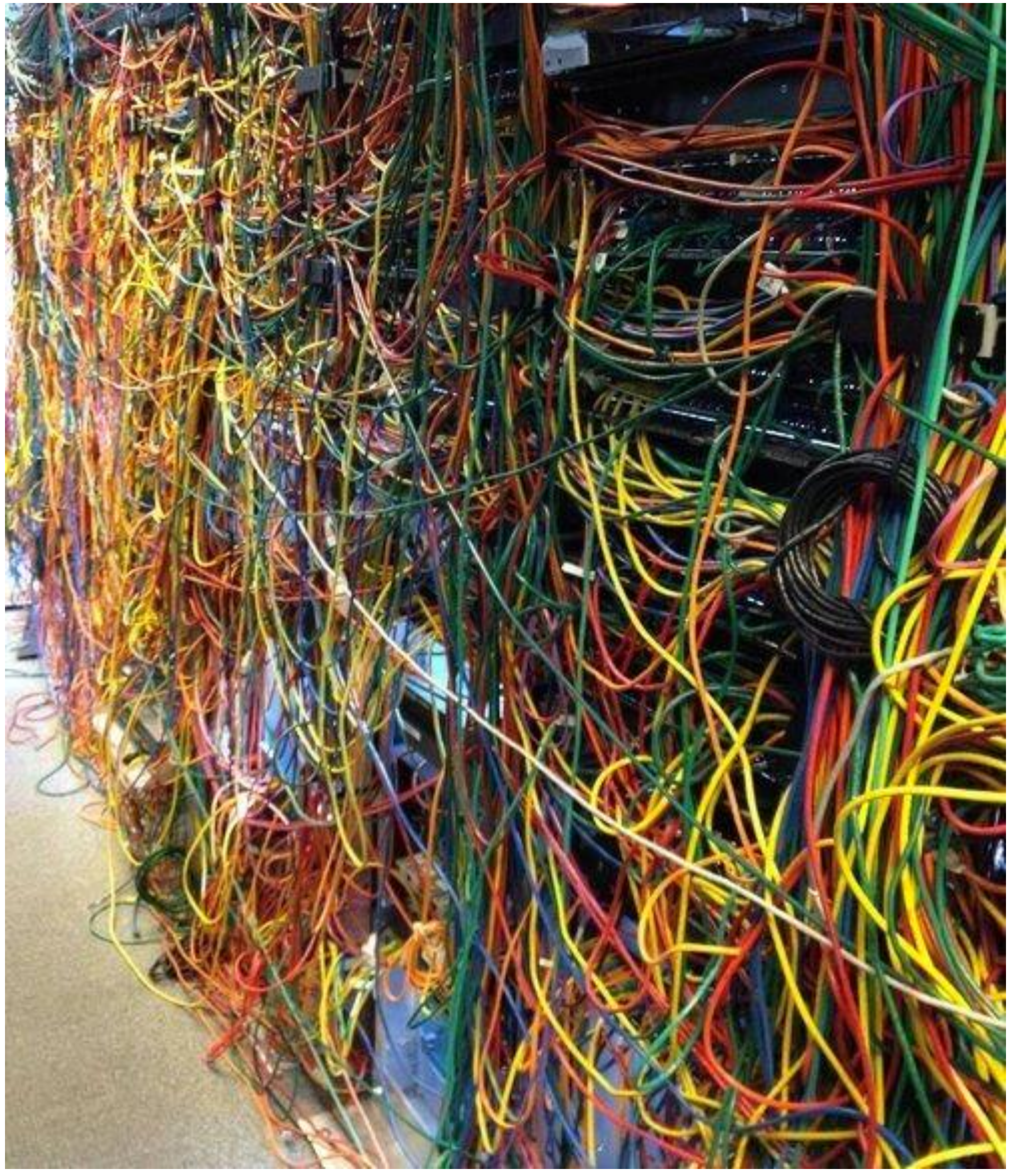
- is a python module that makes data science easy and effective.

Example questions related to datasets

1. What was the max temperature in New York for the month of January?
2. On which days did it rains?
3. What was the average wind speed during the month?

Data munging or Data wrangling

- Process of cleaning messy data



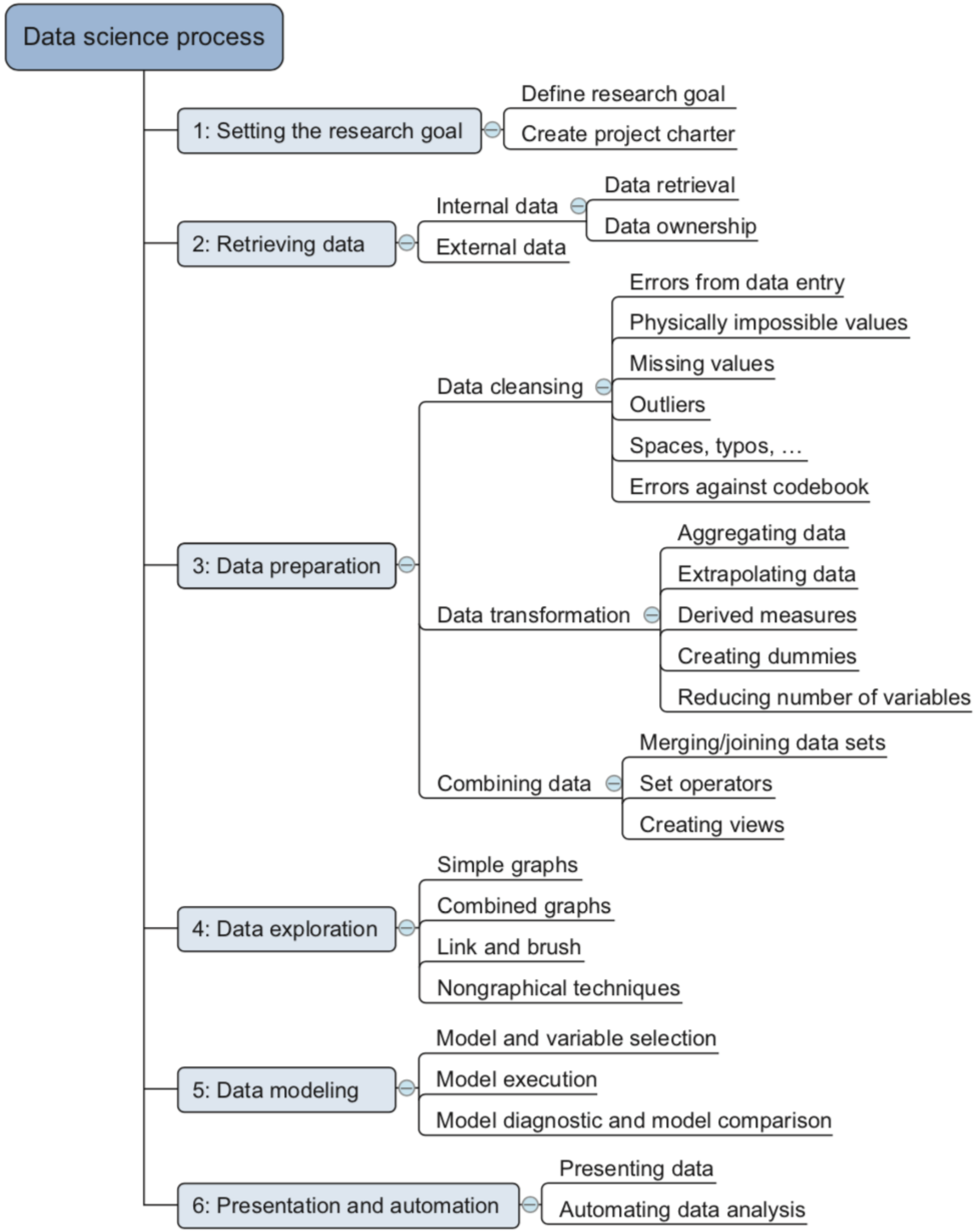
RAW DATA

CLEAN DATA

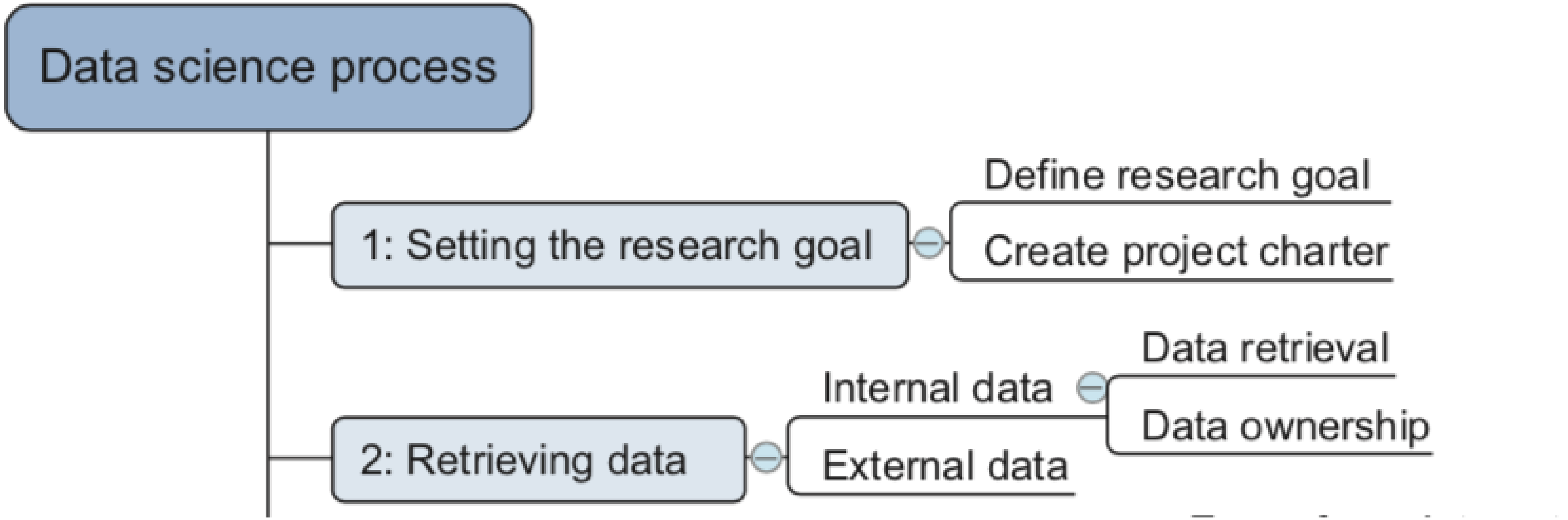
Objective

- Understanding the flow of a data science process
- Discussing the steps in a data science process

Overview


Data science process
1: Setting the research goal
Define research goal
Create project charter
2: Retrieving data
Internal data
Data retrieval
Data ownership
External data
3: Data preparation
Data cleansing
Errors from data entry
Physically impossible values
Missing values
Outliers
Spaces, typos, …
Errors against codebook
Data transformation
Aggregating data
Extrapolating data
Derived measures
Creating dummies
Reducing number of variables
Combining data
Merging/joining data sets
Set operators
Creating views
4: Data exploration
Simple graphs
Combined graphs
Link and brush
Nongraphical techniques
5: Data modeling
Model and variable selection
Model execution
Model diagnostic and model comparison
6: Presentation and automation
Presenting data
Automating data analysis

Overview



Data science process

1: Setting the research goal

Define research goal

Create project charter

2: Retrieving data

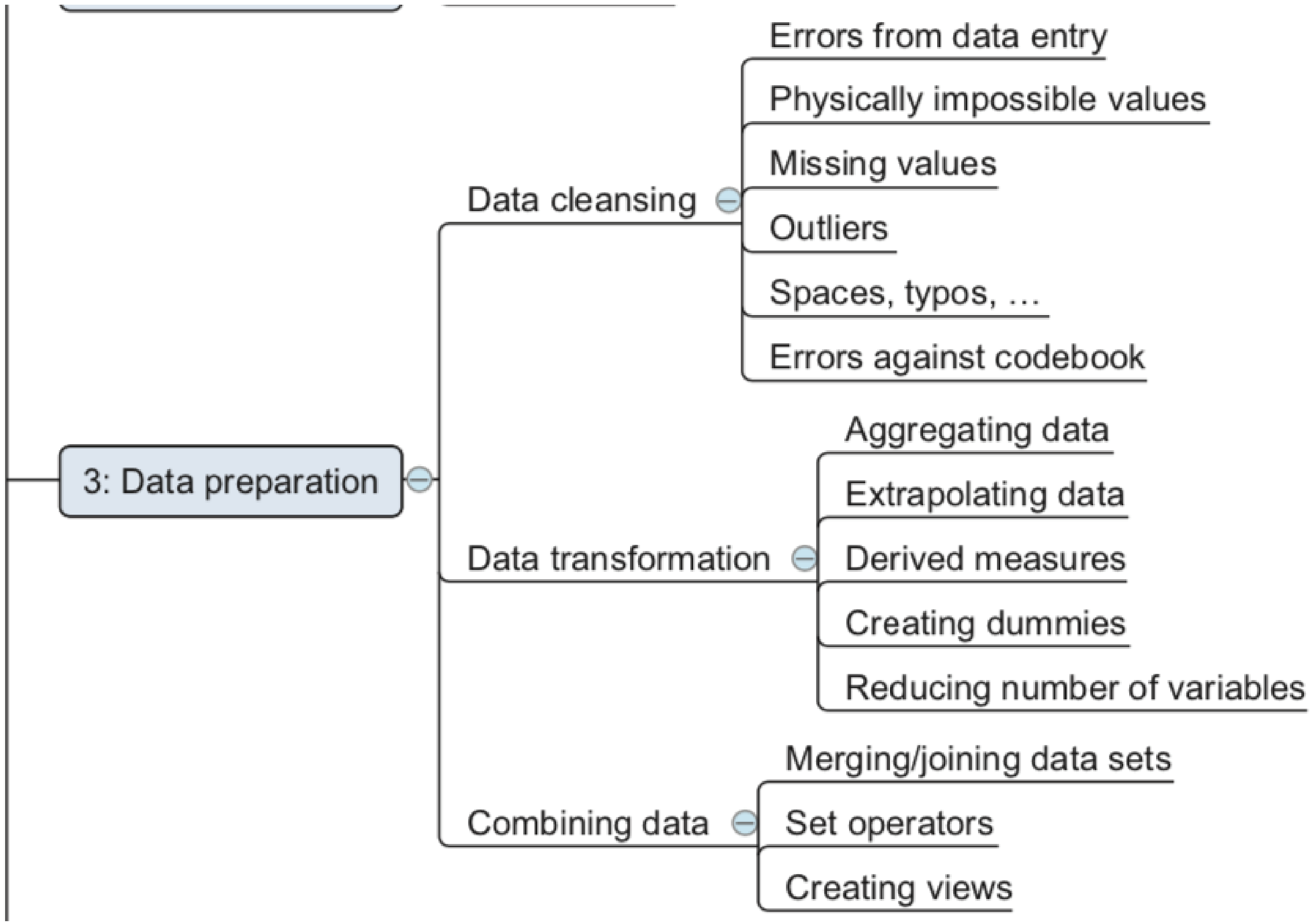
Internal data

Data retrieval

Data ownership

External data

Overview



3: Data preparation

Data cleansing

- Errors from data entry
- Physically impossible values
- Missing values
- Outliers
- Spaces, typos, …
- Errors against codebook

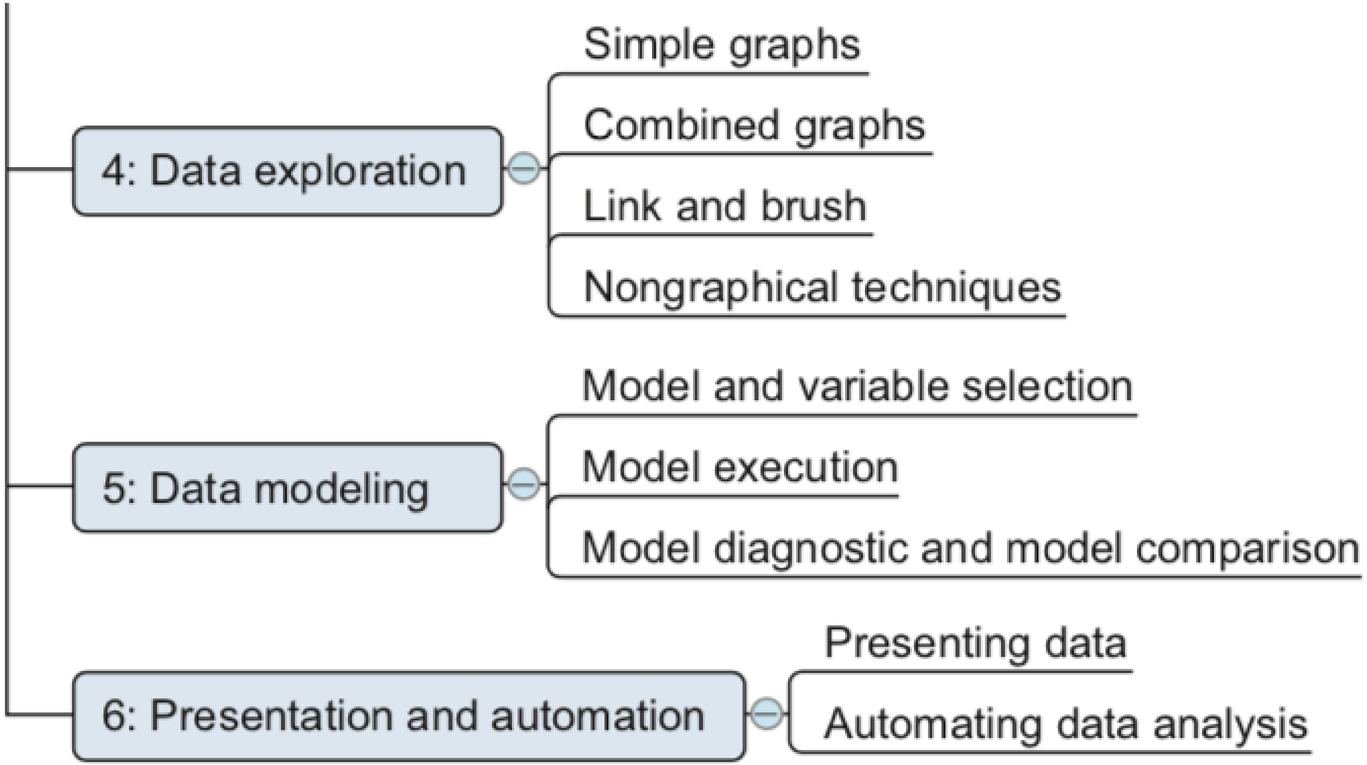
Data transformation

- Aggregating data
- Extrapolating data
- Derived measures
- Creating dummies
- Reducing number of variables

Combining data

- Merging/joining data sets
- Set operators
- Creating views

Overview



4: Data exploration

- Simple graphs
- Combined graphs
- Link and brush
- Nongraphical techniques

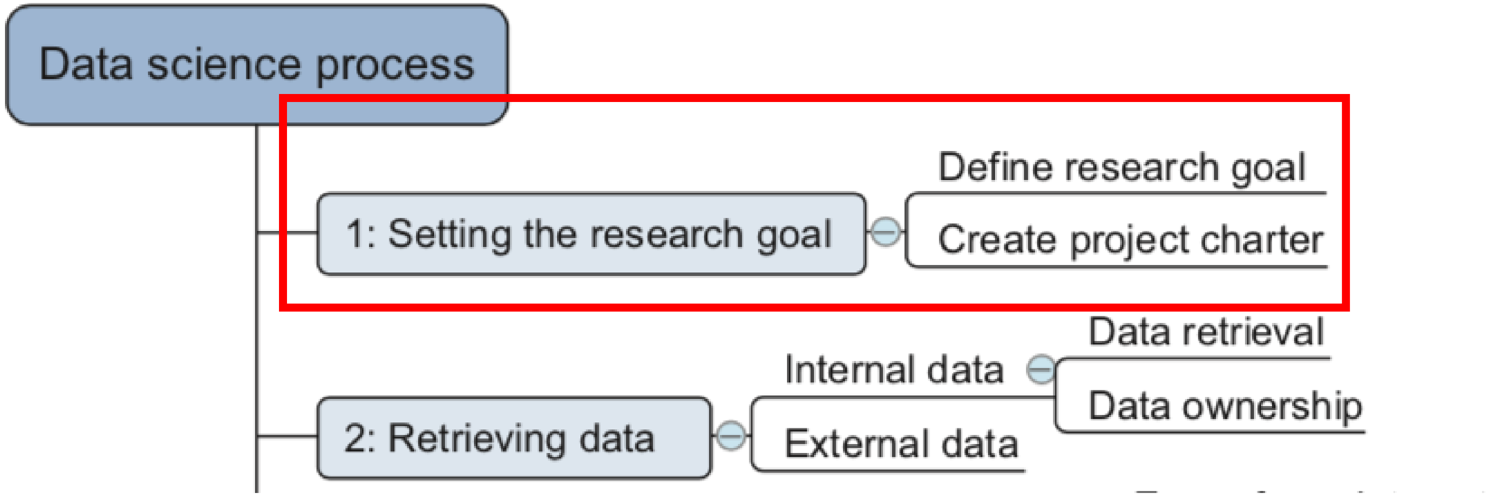
5: Data modeling

- Model and variable selection
- Model execution
- Model diagnostic and model comparison

6: Presentation and automation

- Presenting data
- Automating data analysis

Step 1: Defining research goals and creating a project charter



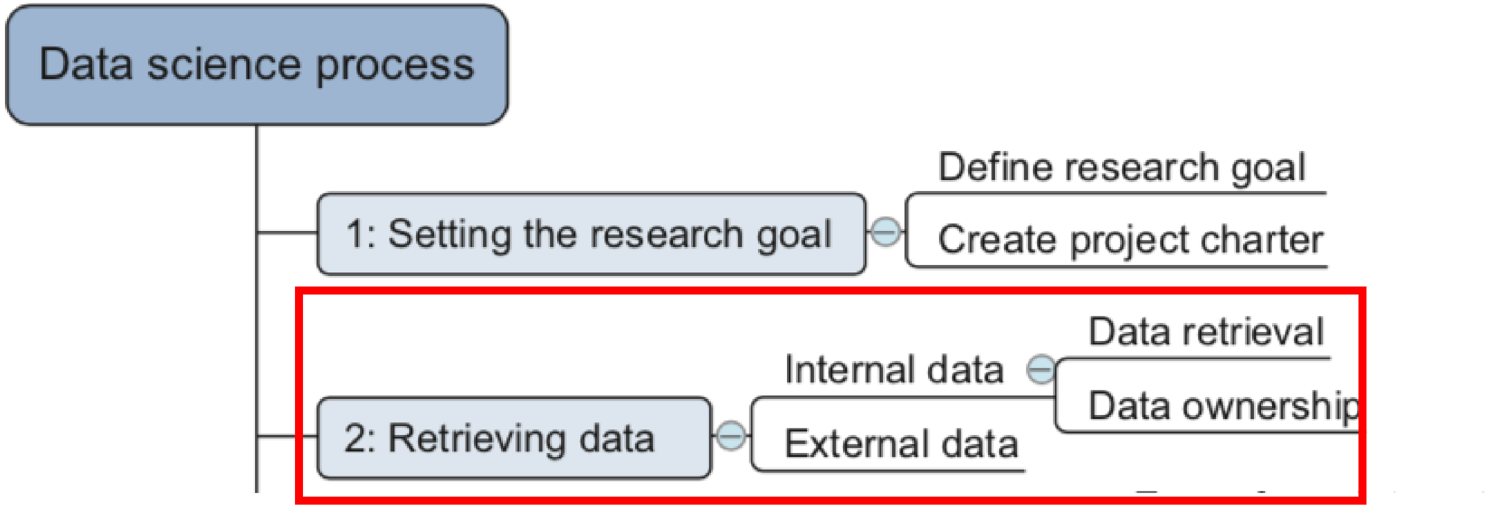
Step 1: Defining research goals and creating a project charter

- Understanding the *what*, the *why*, and the *how* of your project
- What does the company expect you to do?
- Why does management place such a value on your research?
- Is it part of a bigger strategic picture or a “lone wolf” project originating from an opportunity someone detected?
- Outcome
  - clear research goal,
  - good understanding of the context,
  - well-defined deliverables, and a plan of action with a timetable.

Step 1: Defining research goals and creating a project charter

- *Spend time understanding the goals and context of your research*
- 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



Step 2: Retrieving data

- Objective – acquire data researcher needs
- Data can be stored in many forms (text files and tables in db)
- Data is often like a diamond in the rough: it needs polishing to be of any use to you.

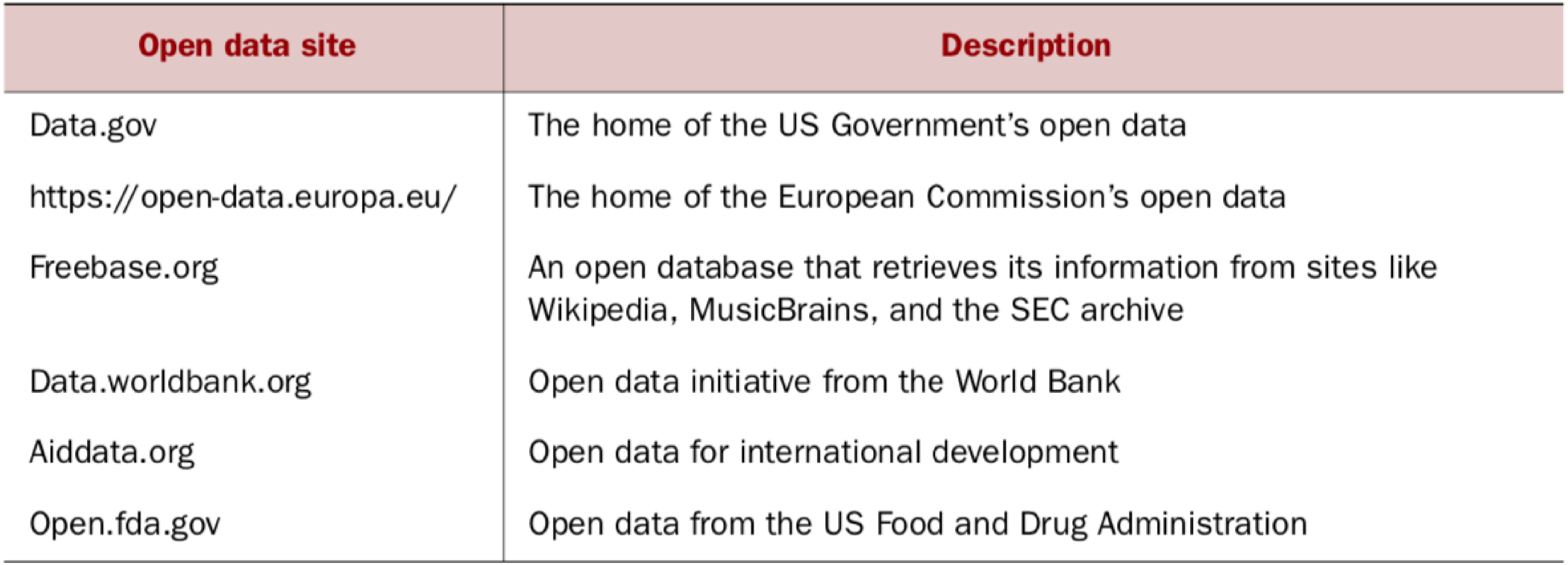
Step 2: Retrieving data

- Start with data stored within the company
  - assess the relevance and quality of the data that’s readily available within your company.
  - Database – data storage
  - Data warehouses - for reading and analyzing that data
  - Data marts - subset of the data warehouse and geared toward serving a specific business unit
  - Data lakes - data in its natural or raw format
  - Excel files on the desktop of a domain expert
- Getting access to data is a difficult task
  - Value and sensitivity of the data

Step 2: Retrieving data

- Don’t be afraid to shop around
  - If data isn’t available inside your organization, look outside your organization’s walls.
  - Governments and organizations share their data for free with the world
  - This data can be of excellent quality; it depends on the institution that creates and manages it.

Step 2: Retrieving data

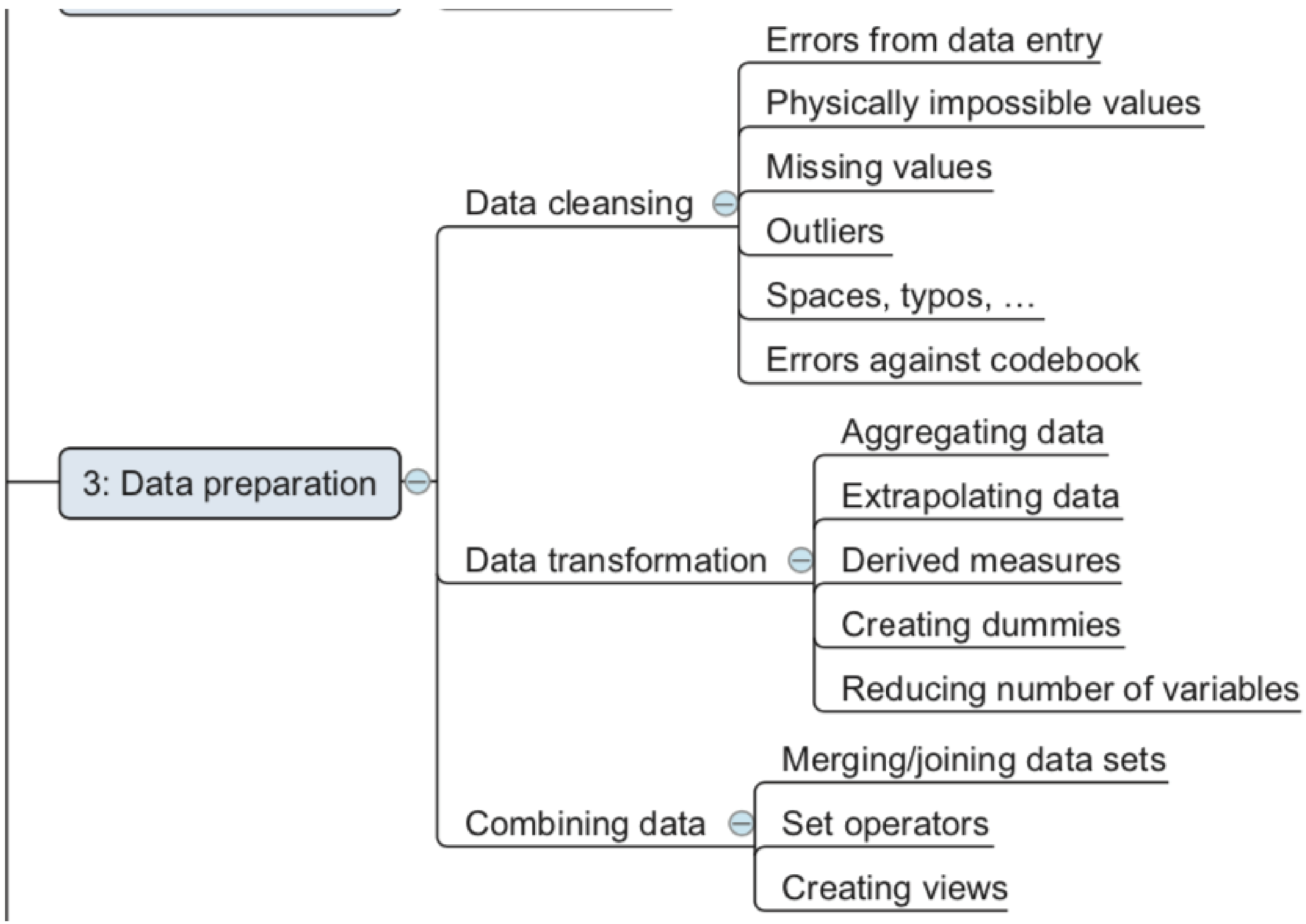


Open data site	Description
Data.gov	The home of the US Government's open data
https://open-data.europa.eu/	The home of the European Commission's open data
Freebase.org	An open database that retrieves its information from sites like Wikipedia, MusicBrains, and the SEC archive
Data.worldbank.org	Open data 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 2: Retrieving data

- Do data quality checks now to prevent problems later
  - Expect to spend a good portion of your project time doing data correction and cleansing up to 80%.
  - *data retrieval*, you check to see if the data is equal to the data in the source document and look to see if you have the right data types.
  - *data preparation*, you do a more elaborate check.
  - The focus is on the content of the variables: you want to get rid of typos and other data entry errors and bring the data to a common standard among the data sets.

Step 3: Cleansing, integrating, and transforming data

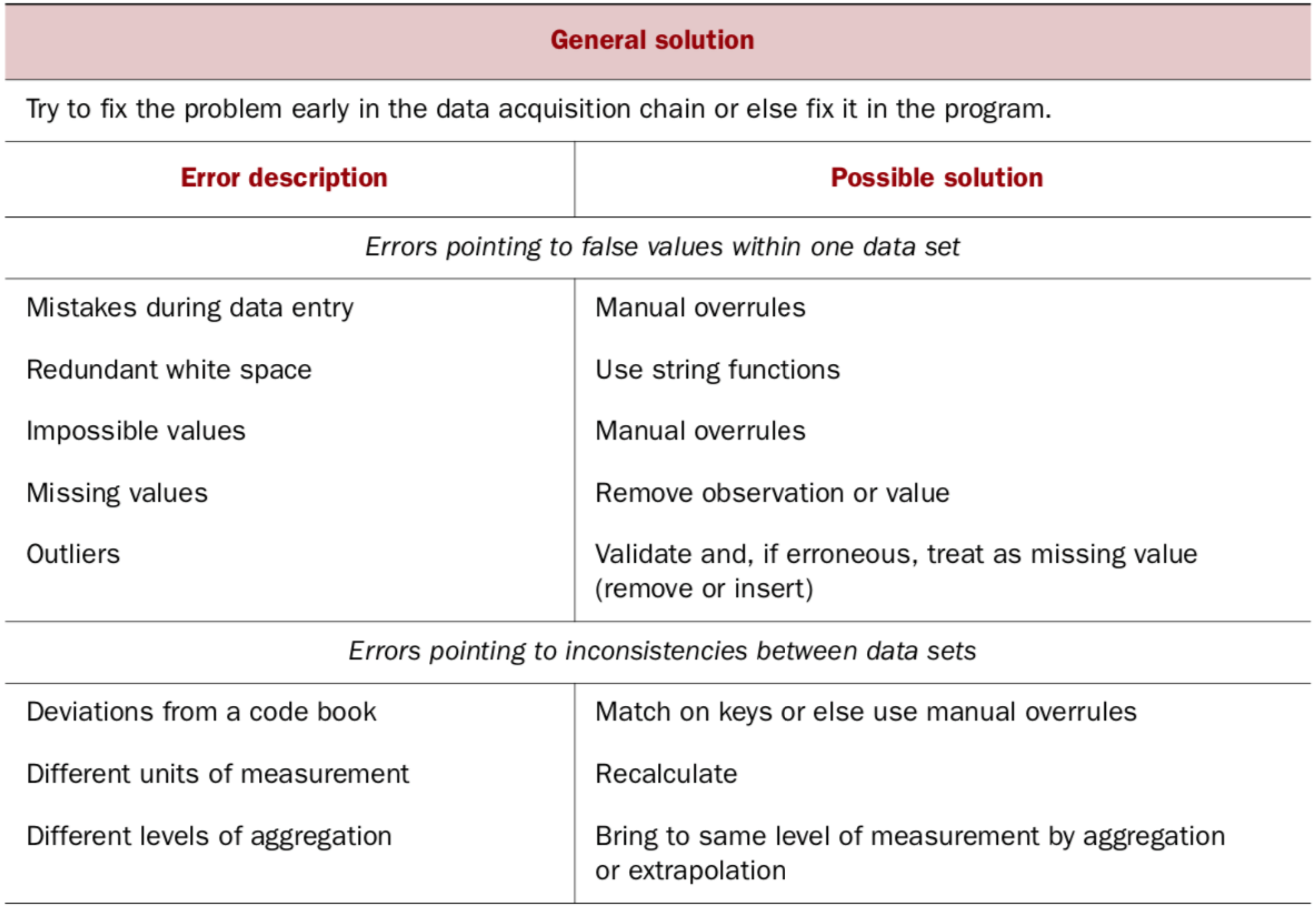


Step 3: Cleansing, integrating, and transforming data

- Cleansing data
  - Data cleansing is a subprocess of the data science process that focuses on removing errors in your data
  - Two types of error
    - *interpretation error (e.g.* person’s age is greater than 300 years*)*
    - *inconsistencies (e.g.* “Female” in one table and “F”*)*

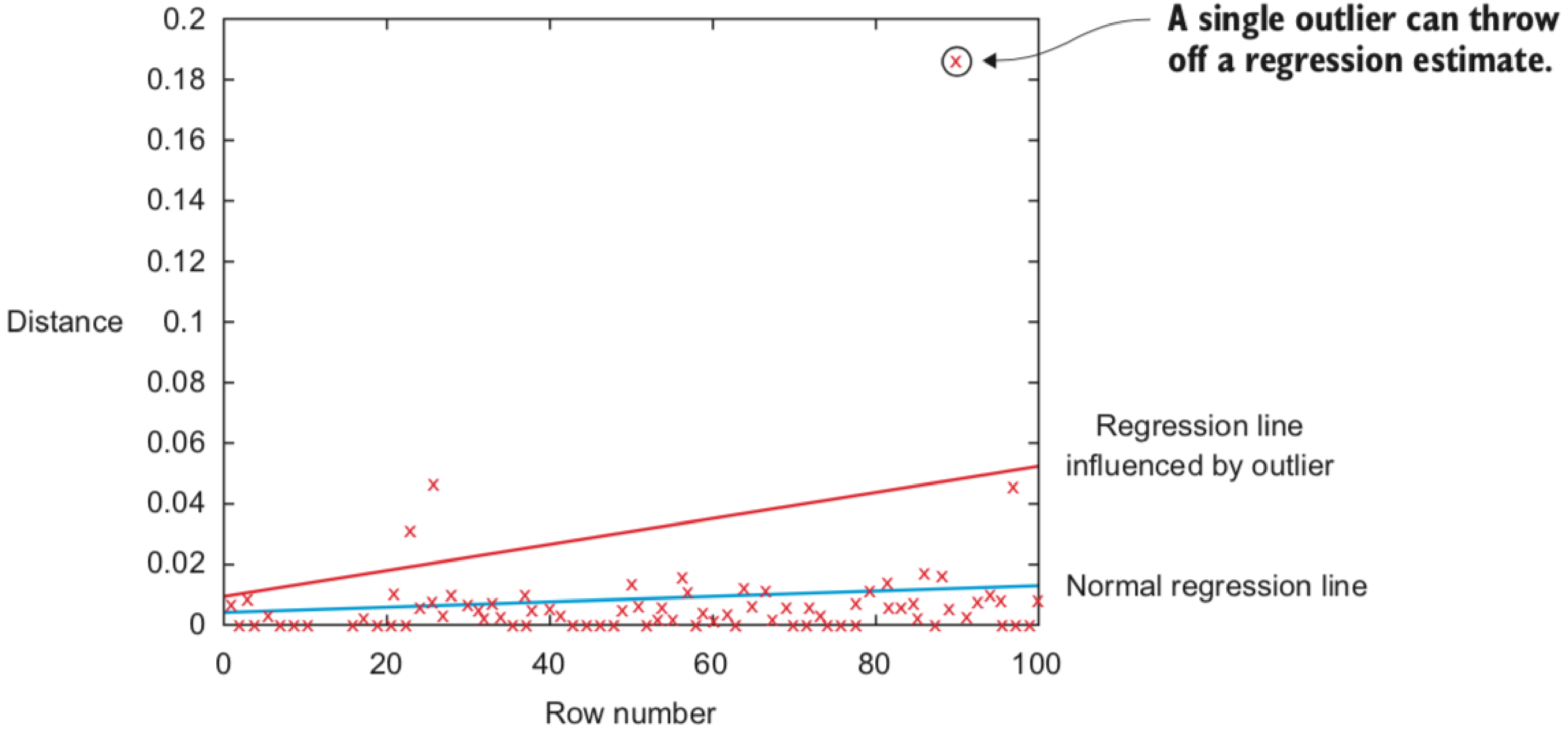
Step 3: Cleansing, integrating, and transforming data

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

Step 3: Cleansing, integrating, and transforming data

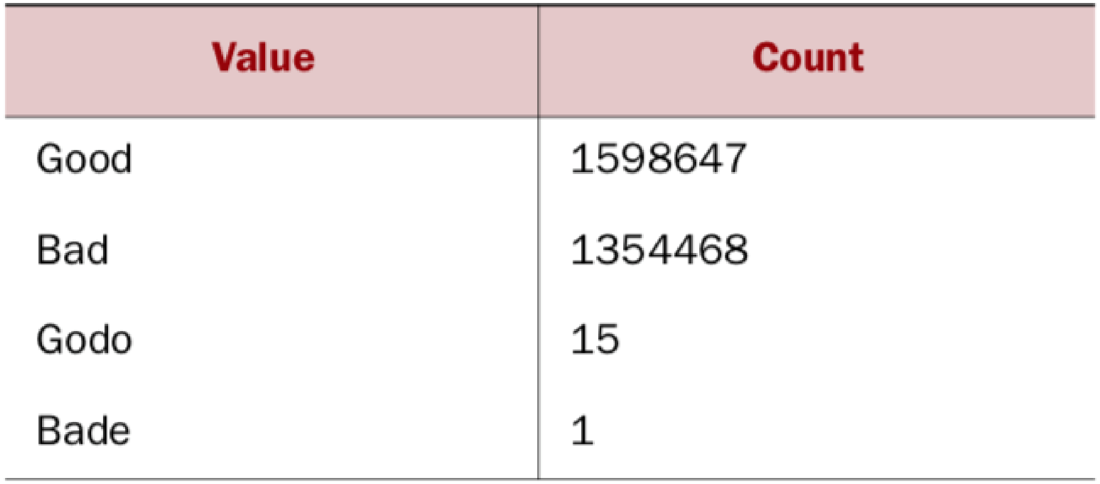


A single outlier can throw off a regression estimate.

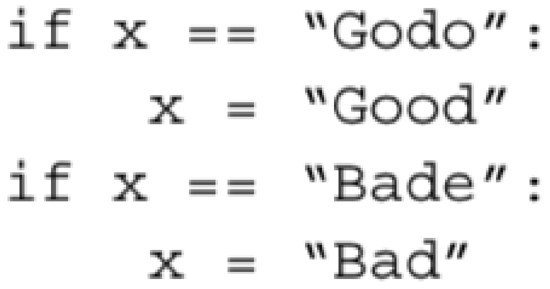
The encircled point influences the model heavily and is worth investigating because it can point to a region where you don’t have enough data or might indicate an error in the data, but it also can be a valid data point.

Step 3: Cleansing, integrating, and transforming data

- Data Entry Errors
  - Data collection and data entry are error-prone processes.
  - Errors can arise from human sloppiness, whereas others are due to machine or hardware failure.
  - When you have a variable that can take only two values: “Good” and “Bad”, you can create a frequency table and see if those are truly the only two values present.



Value	Count
Good	1598647
Bad	1354468
Godo	15
Bade	1



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

Step 3: Cleansing, integrating, and transforming data

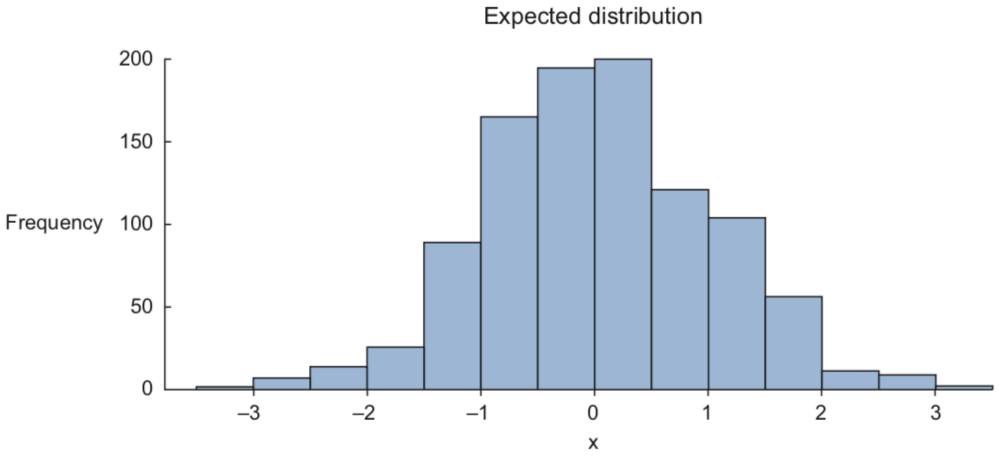
- Redundant Whitespace
  - Whitespaces tend to be hard to detect but cause errors like other redundant characters would.
  - If you know to watch out for them, fixing redundant whitespaces is luckily easy enough in most programming languages.
  - Python you can use the *strip()* function to remove leading and trailing spaces.
- 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.

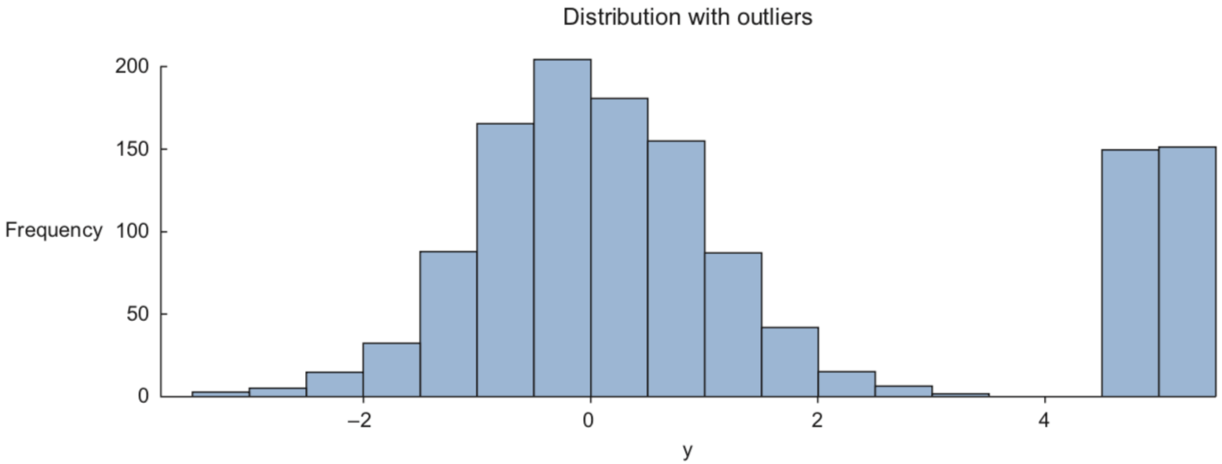
Step 3: Cleansing, integrating, and transforming data

- Impossible Values And Sanity Checks
    - 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*

Step 3: Cleansing, integrating, and transforming data

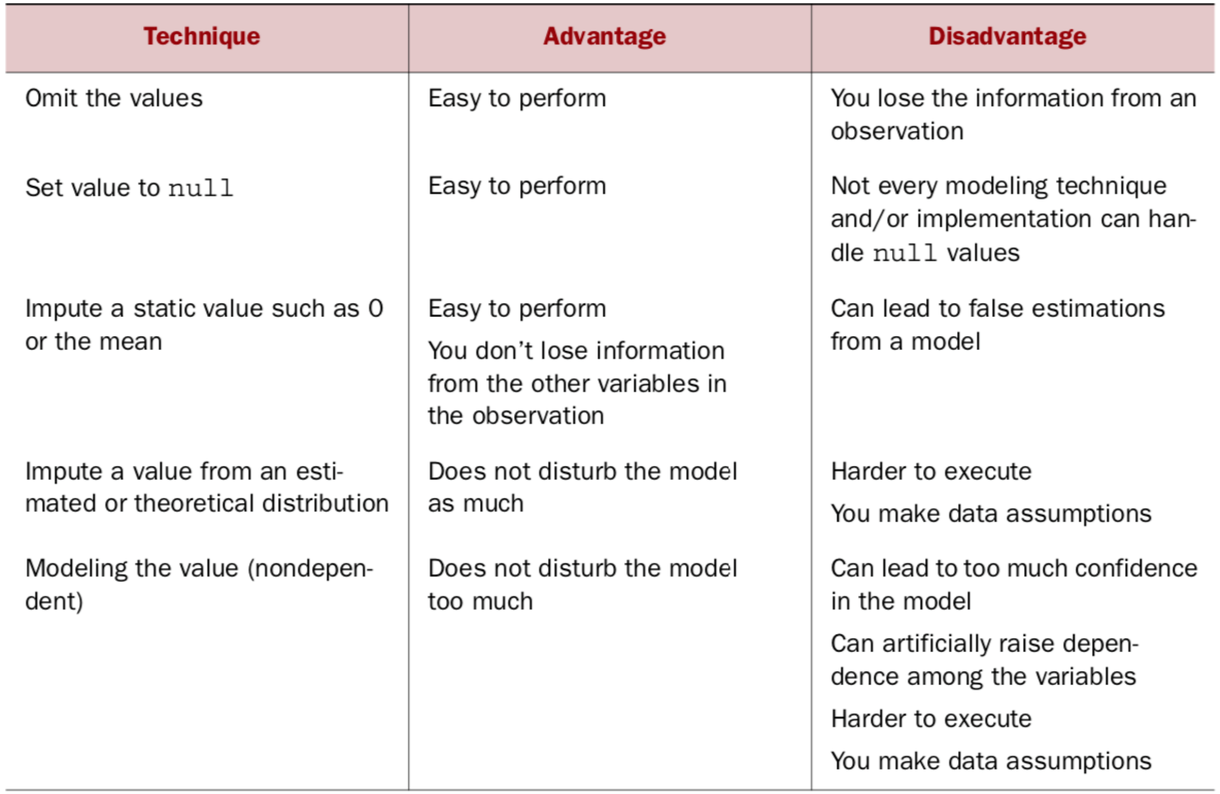
- Outliers
  - The easiest way to find outliers is to use a plot or a table with the minimum and maximum values.
  - As we saw earlier with the regression, outliers can gravely influence your data modeling, so investigate them first.





Step 3: Cleansing, integrating, and transforming data

- Dealing With Missing Values
  - Missing values aren’t necessarily wrong, but you still need to handle them separately;
  - An overview of techniques to handle missing data (see table)



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

Step 3: Cleansing, integrating, and transforming data

- Deviations From A Code Book
  - A code book is a description of your data, a form of metadata.
  - It contains things such as the number of variables per observation, the number of observations, and what each encoding within a variable means. (For instance “0” equals “negative”, “5” stands for “very positive”.)
- Different Units Of Measurement
  - When integrating two data sets, you have to pay attention to their respective units of measurement.
  - A simple conversion will do the trick in this case.

Step 3: Cleansing, integrating, and transforming data

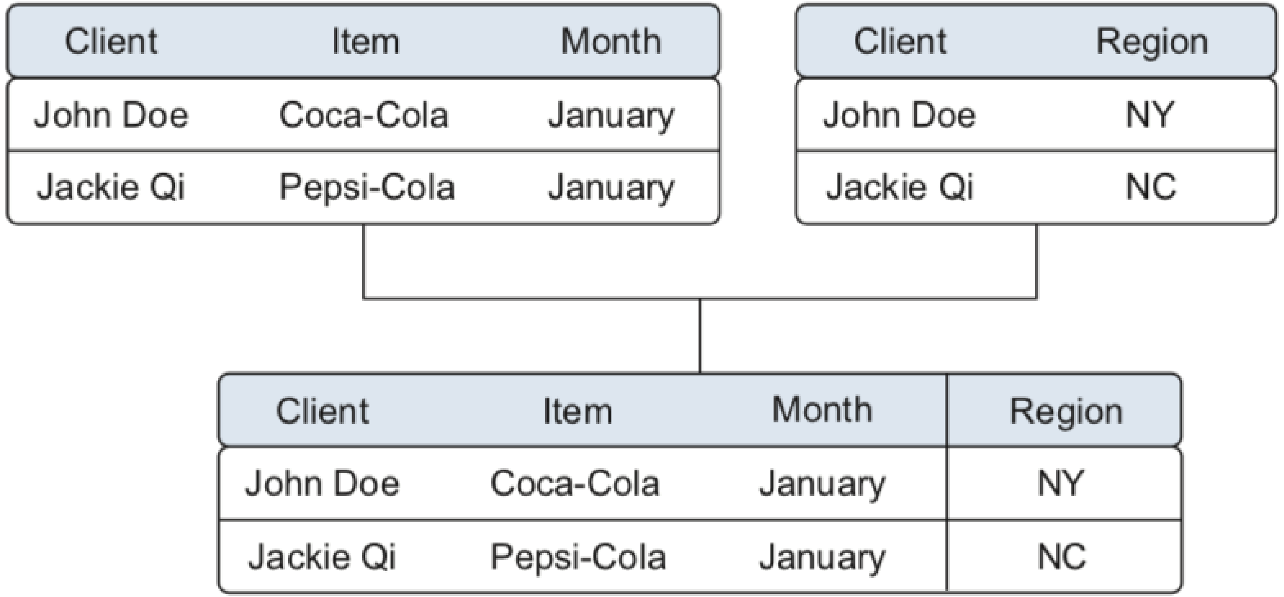
- Correct errors as early as possible
  - A good practice is to mediate data errors as early as possible in the data collection chain and to fix as little as possible inside your program while fixing the origin of the problem.
  - If you can’t correct the data at the source, you’ll need to handle it inside your code.
  - Always keep a copy of your original data (if possible)
- Combining data from different data sources
  - Your data comes from several different places, and in this substep we focus on integrating these different sources.
  - Data varies in size, type, and structure, ranging from databases and Excel files to text documents.

Step 3: Cleansing, integrating, and transforming data

- The Different Ways Of Combining Data
  - The first operation is *joining*: enriching an observation from one table with information from another table.
  - The second operation is *appending or stacking*: adding the observations of one table to those of another table.

Step 3: Cleansing, integrating, and transforming data

- Joining tables
  - Joining tables allows you to combine the information of one observation found in one table with the information that you find in another table.
  - Joining two tables on the Item and Region keys (see example figure)



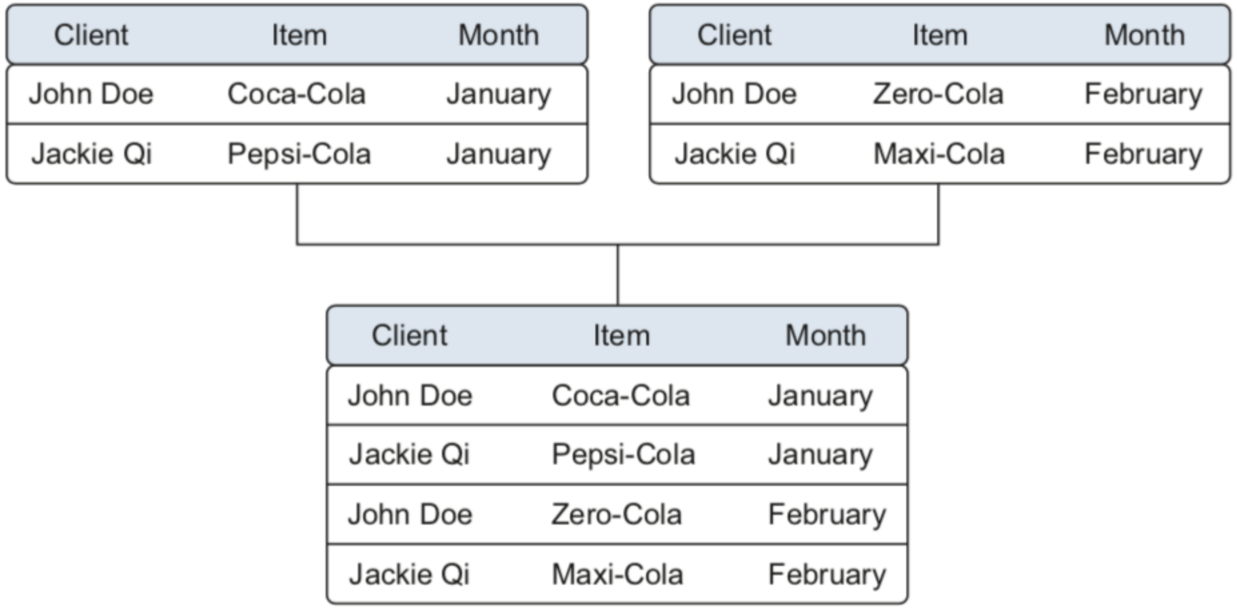
Client	Item	Month
John Doe	Coca-Cola	January
Jackie Qi	Pepsi-Cola	January

Client	Region
John Doe	NY
Jackie Qi	NC

Client	Item	Month	Region
John Doe	Coca-Cola	January	NY
Jackie Qi	Pepsi-Cola	January	NC

Step 3: Cleansing, integrating, and transforming data

- Appending tables
  - Appending or stacking tables is effectively adding observations from one table to another table.
  - Appending data from tables is a common operation but requires an equal structure in the tables being appended. (see example figure)



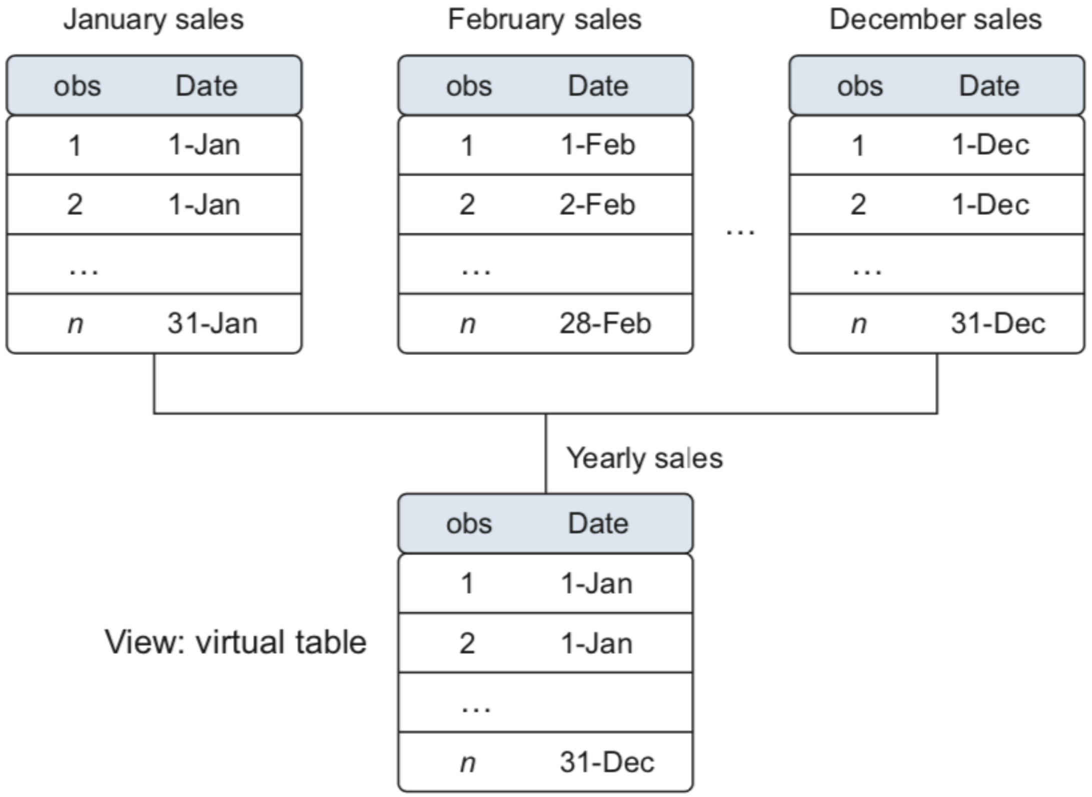
Client	Item	Month
John Doe	Coca-Cola	January
Jackie Qi	Pepsi-Cola	January

Client	Item	Month
John Doe	Zero-Cola	February
Jackie Qi	Maxi-Cola	February

Client	Item	Month
John Doe	Coca-Cola	January
Jackie Qi	Pepsi-Cola	January
John Doe	Zero-Cola	February
Jackie Qi	Maxi-Cola	February

Step 3: Cleansing, integrating, and transforming data

- Using Views To Simulate Data Joins And Appends
  - To avoid duplication of data, you virtually combine data with views.
  - The problem is that we duplicated the data and therefore needed more storage space.
  - A view behaves as if you’re working on a table, but this table is nothing but a virtual layer that combines the tables for you.
  - Views do come with a draw- back, however. The join that creates the view is recreated every time it’s queried, using more processing power than a pre-calculated table would have.



January sales

obs	Date
1	1-Jan
2	1-Jan
…	
n	31-Jan

February sales

obs	Date
1	1-Feb
2	2-Feb
…	
n	28-Feb

…

December sales

obs	Date
1	1-Dec
2	1-Dec
…	
n	31-Dec

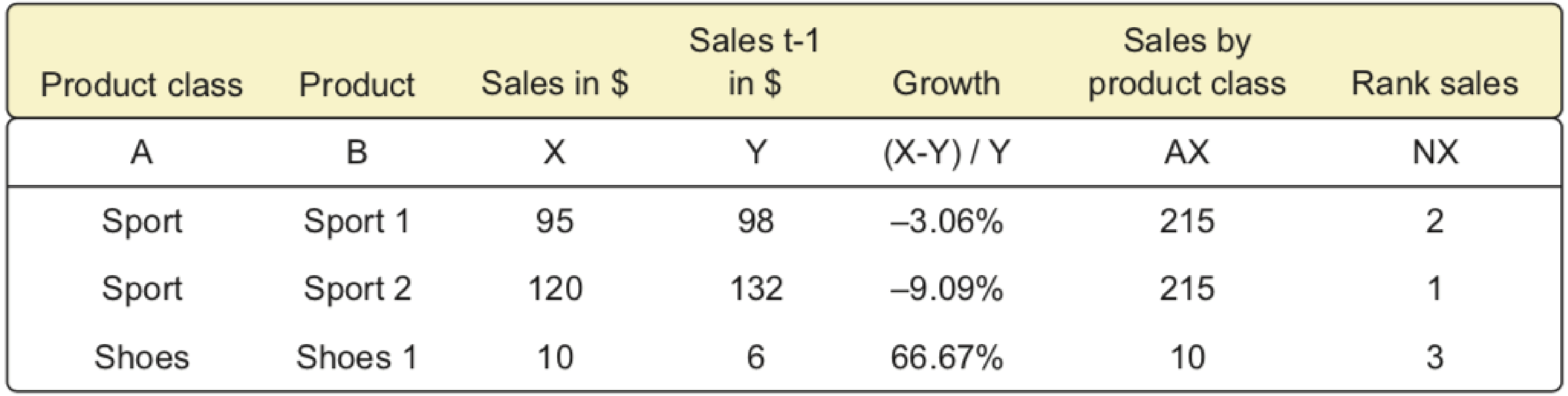
Yearly sales

View: virtual table

obs	Date
1	1-Jan
2	1-Jan
…	
n	31-Dec

Step 3: Cleansing, integrating, and transforming data

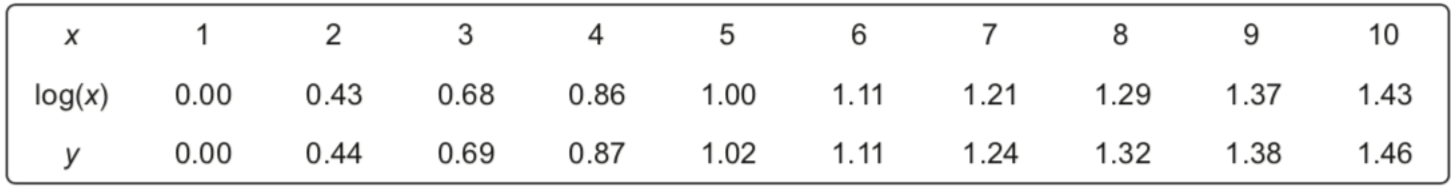
- Enriching Aggregated Measures
  - Data enrichment can also be done by adding calculated information to the table, such as the total number of sales or what percentage of total stock has been sold in a certain region (see example figure)



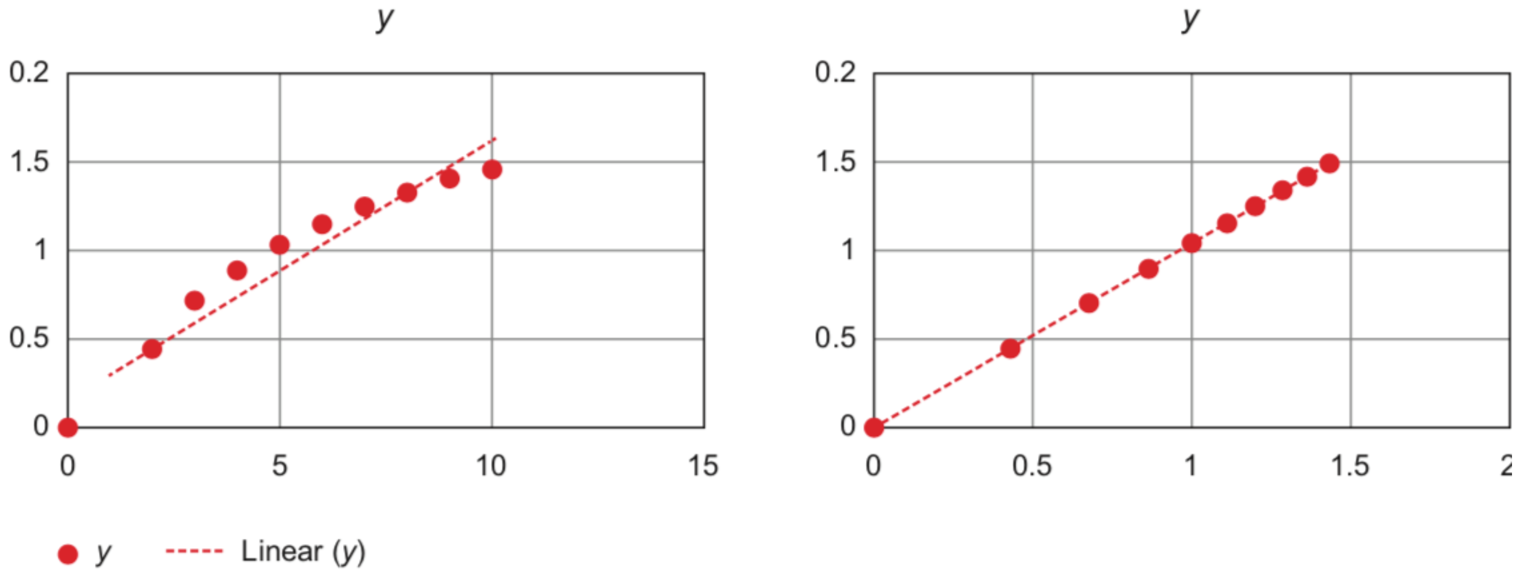
Product class	Product	Sales in $	Sales t-1 in $	Growth	Sales by product class	Rank sales
A	B	X	Y	(X-Y) / Y	AX	NX
Sport	Sport 1	95	98	–3.06%	215	2
Sport	Sport 2	120	132	–9.09%	215	1
Shoes	Shoes 1	10	6	66.67%	10	3

Step 3: Cleansing, integrating, and transforming data

- Transforming data
  - Relationships between an input variable and an output variable aren’t always linear. Take, for instance, a relationship of the form ***y = aebx***. Taking the log of the independent variables simplifies the estimation problem dramatically.



x	1	2	3	4	5	6	7	8	9	10
log(*x*)	0.00	0.43	0.68	0.86	1.00	1.11	1.21	1.29	1.37	1.43
y	0.00	0.44	0.69	0.87	1.02	1.11	1.24	1.32	1.38	1.46

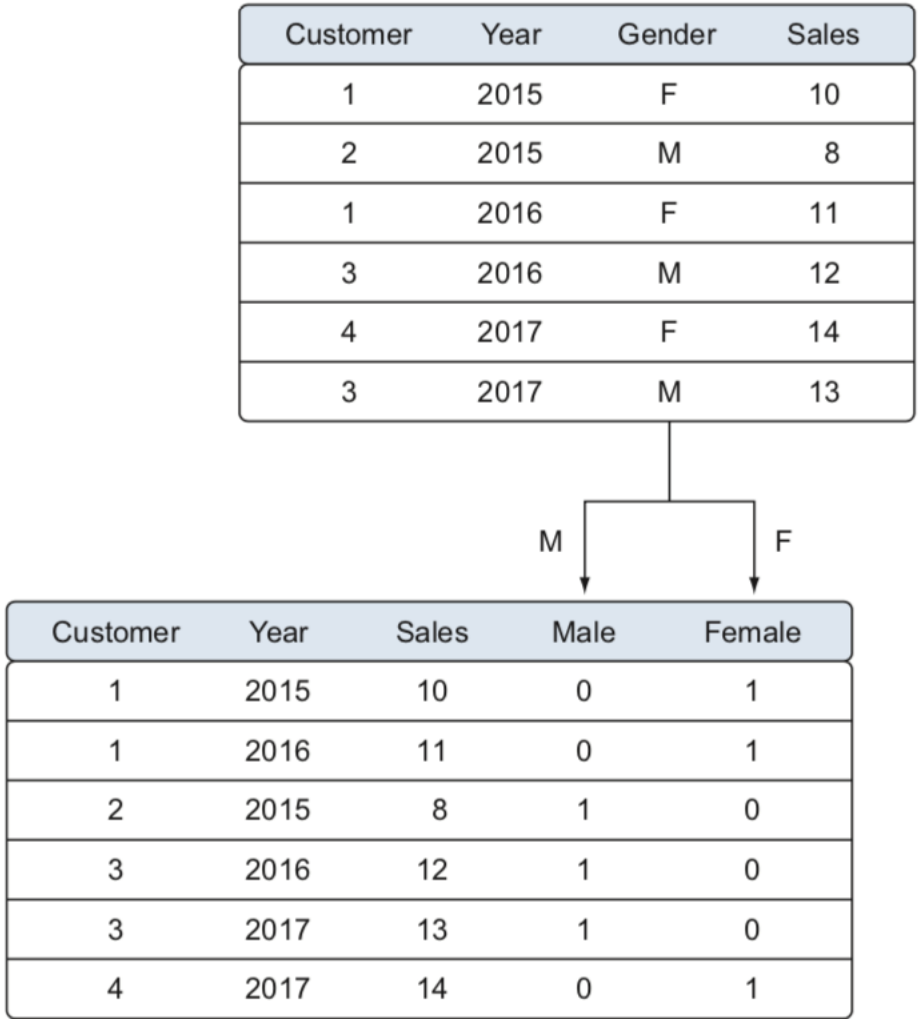


Step 3: Cleansing, integrating, and transforming data

- Reducing The Number Of Variables
  - Sometimes you have too many variables and need to reduce the number because they don’t add new information to the model.
  - Having too many variables in your model makes the model difficult to handle, and certain techniques don’t perform well when you overload them with too many input variables.

Step 3: Cleansing, integrating, and transforming data

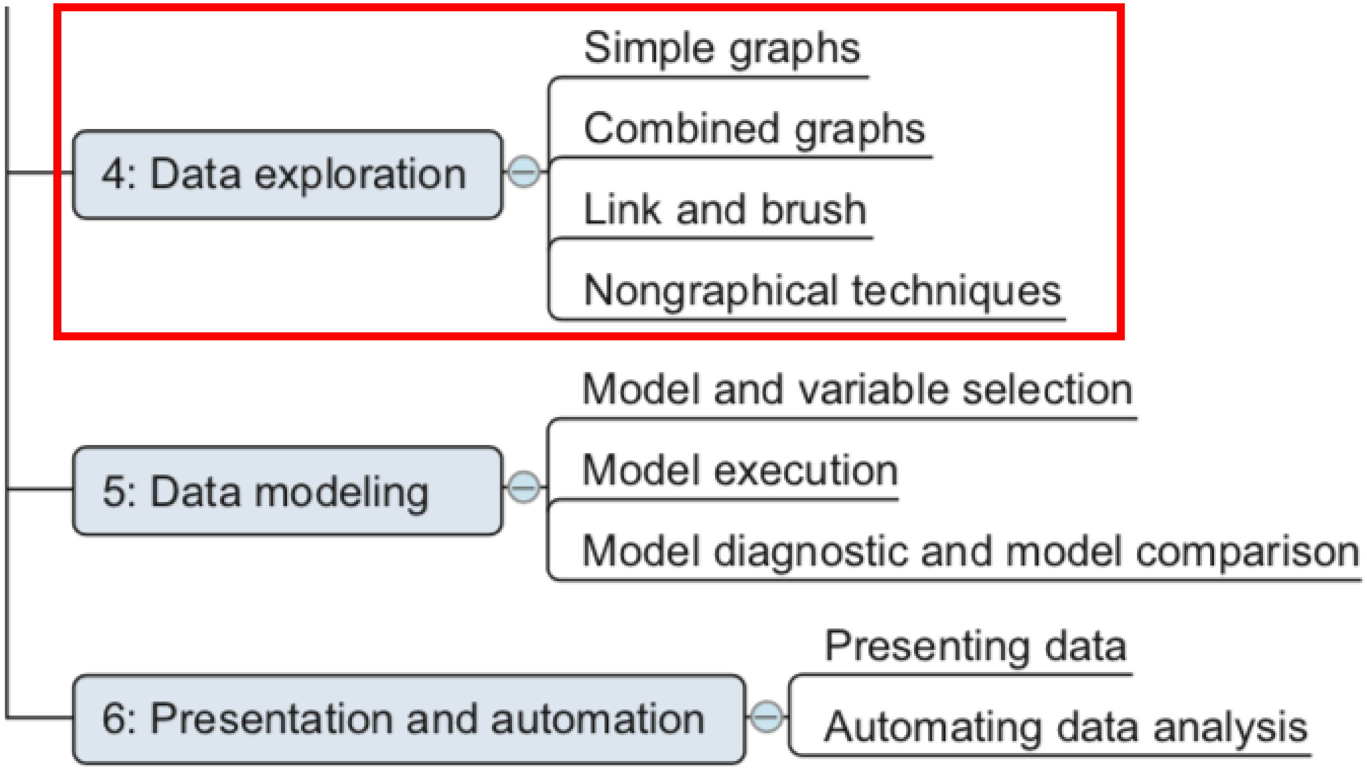
- Turning Variables Into Dummies
  - Dummy variables can only take two values: true(1) or false(0).
  - They’re used to indicate the absence of a categorical effect that may explain the observation.
  - In this case you’ll make separate columns for the classes stored in one variable and indicate it with 1 if the class is present and 0 otherwise.



Customer	Year	Gender	Sales
1	2015	F	10
2	2015	M	8
1	2016	F	11
3	2016	M	12
4	2017	F	14
3	2017	M	13

Customer	Year	Sales	Male	Female
1	2015	10	0	1
1	2016	11	0	1
2	2015	8	1	0
3	2016	12	1	0
3	2017	13	1	0
4	2017	14	0	1

Step 4: Exploratory data analysis



4: Data exploration

- Simple graphs
- Combined graphs
- Link and brush
- Nongraphical techniques

5: Data modeling

- Model and variable selection
- Model execution
- Model diagnostic and model comparison

6: Presentation and automation

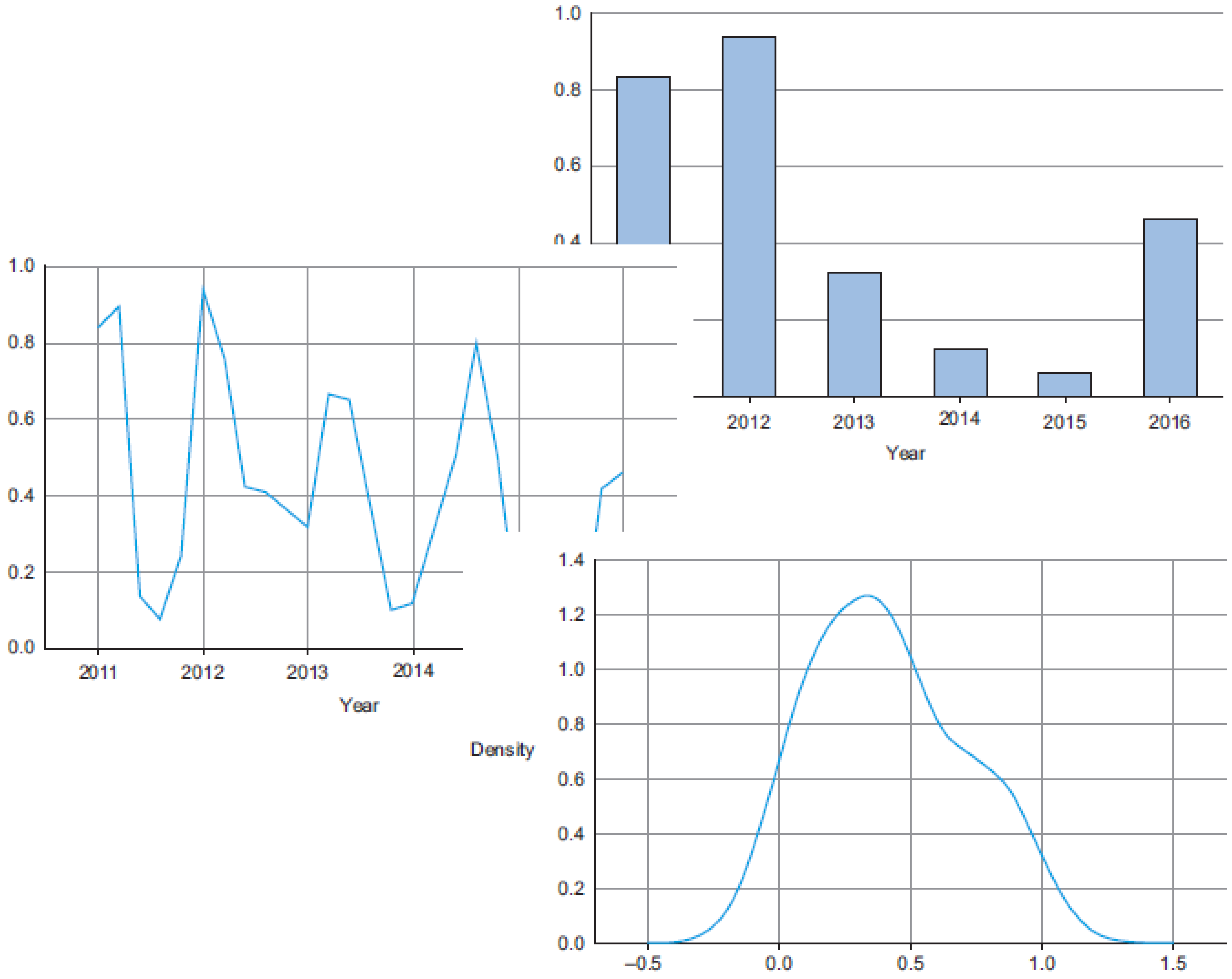
- Presenting data
- Automating data analysis

Step 4: Exploratory data analysis

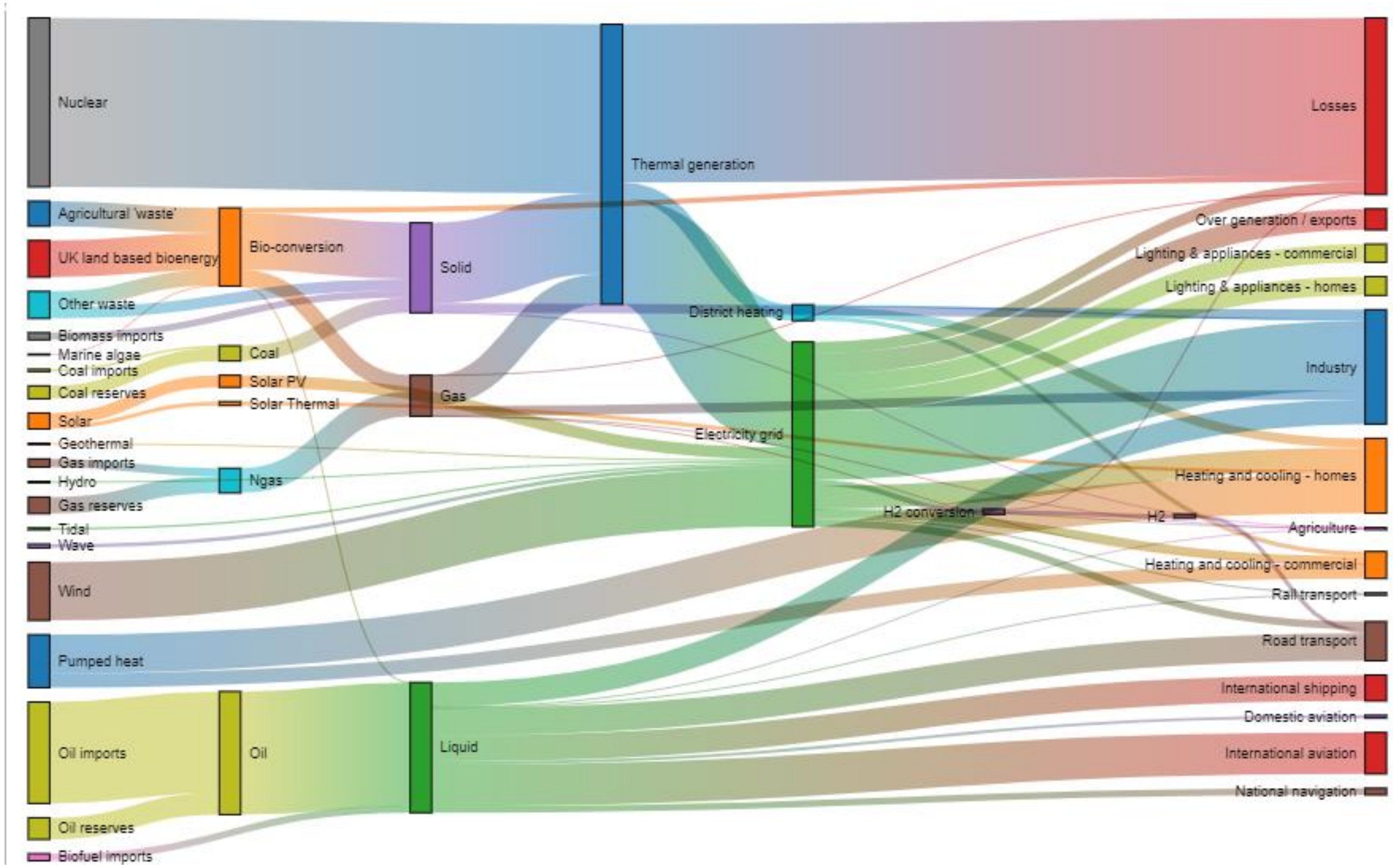
- This phase is about exploring data, so keeping your mind open and your eyes peeled is essential during the exploratory data analysis phase.
- The goal isn’t to cleanse the data, but it’s common that you’ll still discover anomalies you missed before, forcing you to take a step back and fix them.

Step 4: Exploratory data analysis

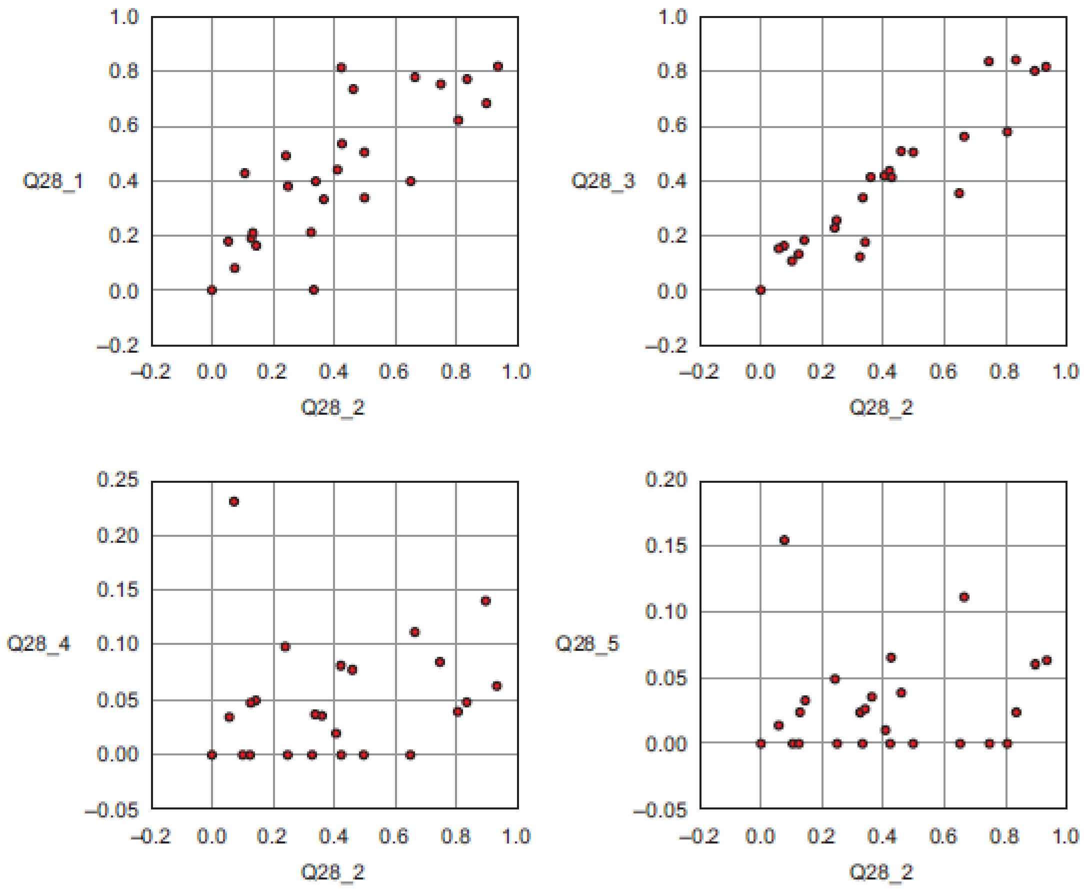
- The visualization techniques you use in this phase range from simple line graphs or histograms, as shown in figure on the right.



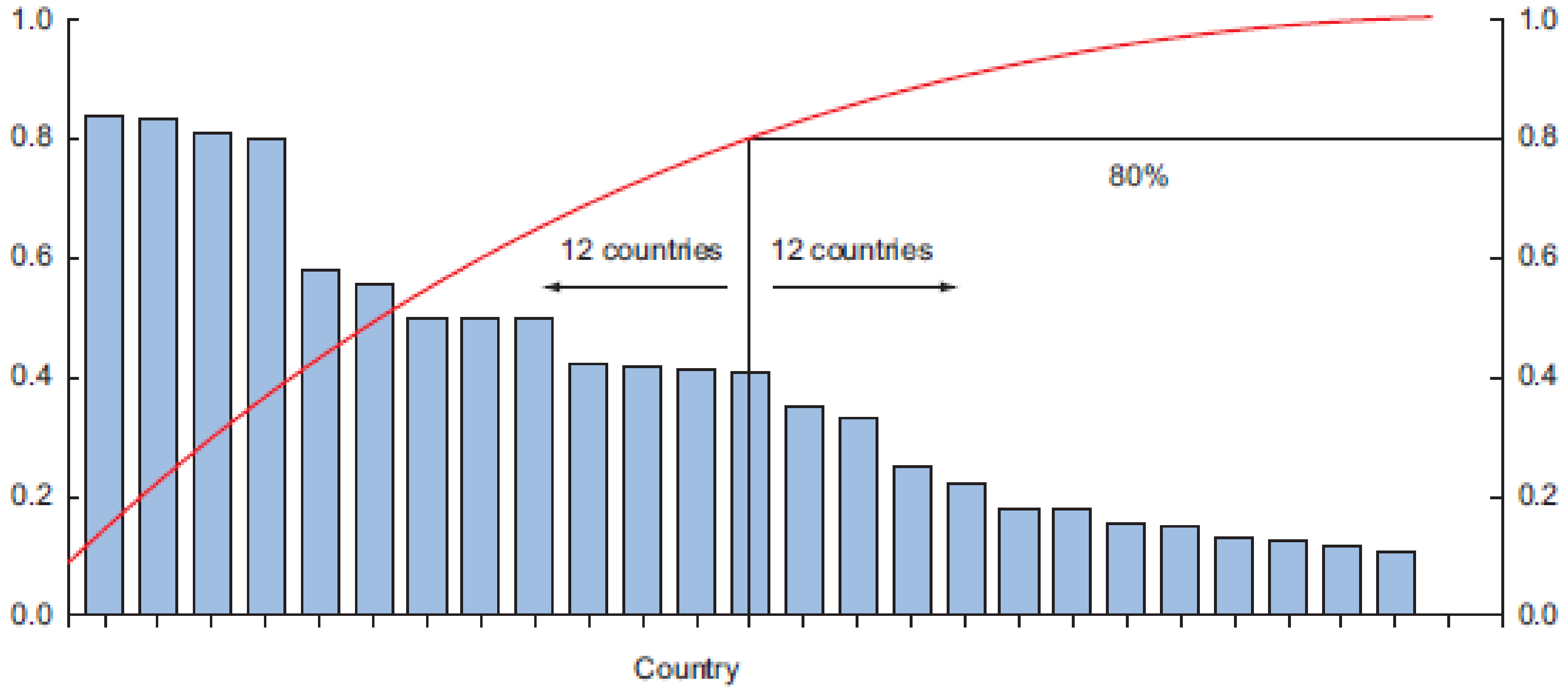
Step 4: Exploratory data analysis



Step 4: Exploratory data analysis

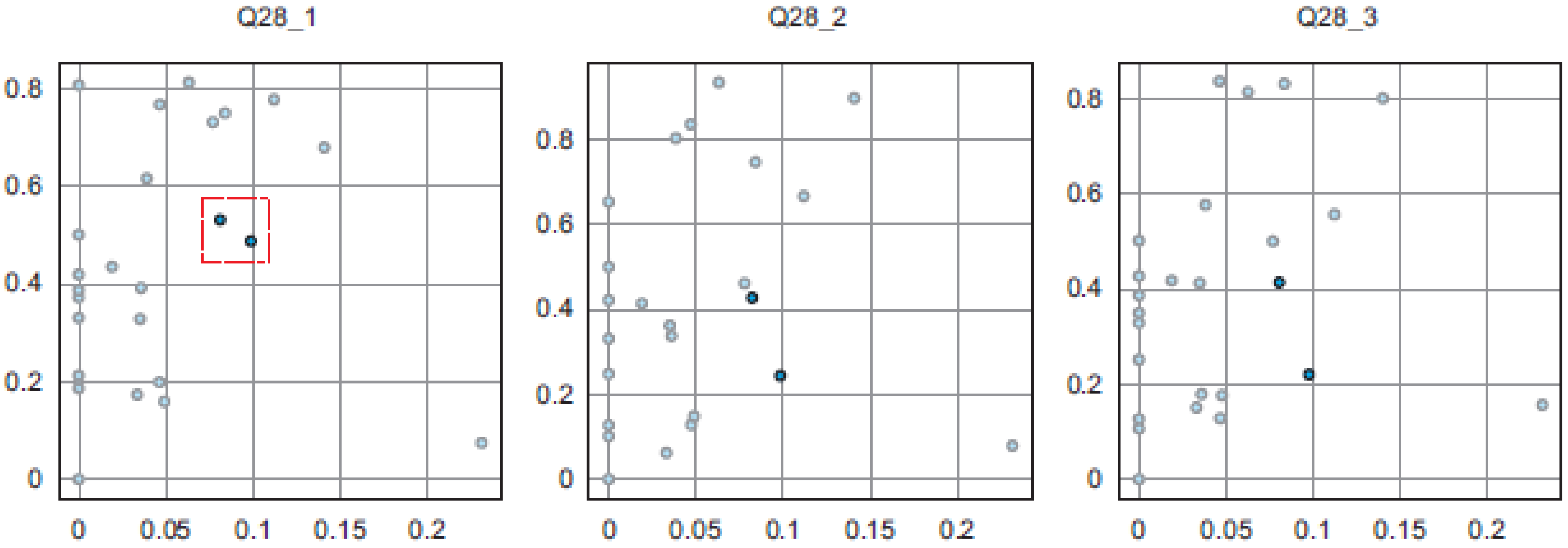


Step 4: Exploratory data analysis



- A Pareto diagram is a combination of the values and a cumulative distribution. It's easy to see from this diagram that the first 50% of the countries contain slightly less than 80% of the total amount.
- If this graph represented customer buying power and we sell expensive products, we probably don't need to spend our marketing budget in every country; we could start with the first 50%.

Step 4: Exploratory data analysis



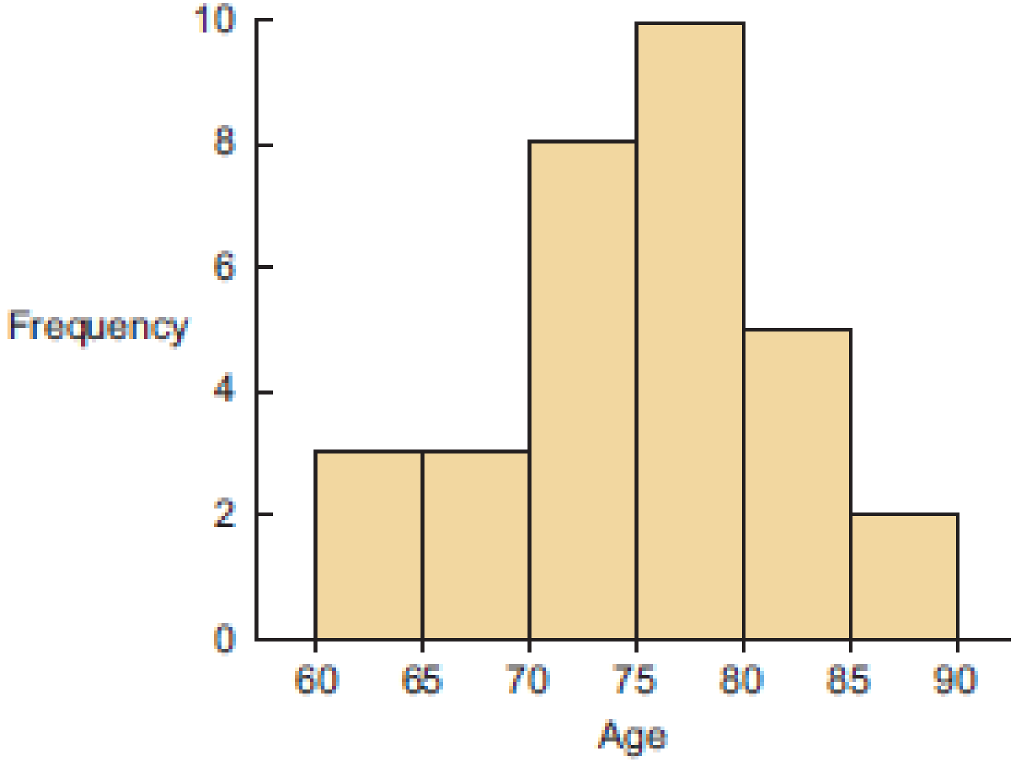
- Link and brush allows you to select observations in one plot and highlight the same observations in the other plots.

Step 4: Exploratory data analysis

- Histogram - a variable is cut into discrete categories and the number of occurrences in each category are summed up and shown in the graph.
- Boxplot - doesn’t show how many observations are present but does offer an impression of the distribution within categories. It can show the maximum, minimum, median, and other characterizing measures at the same time.

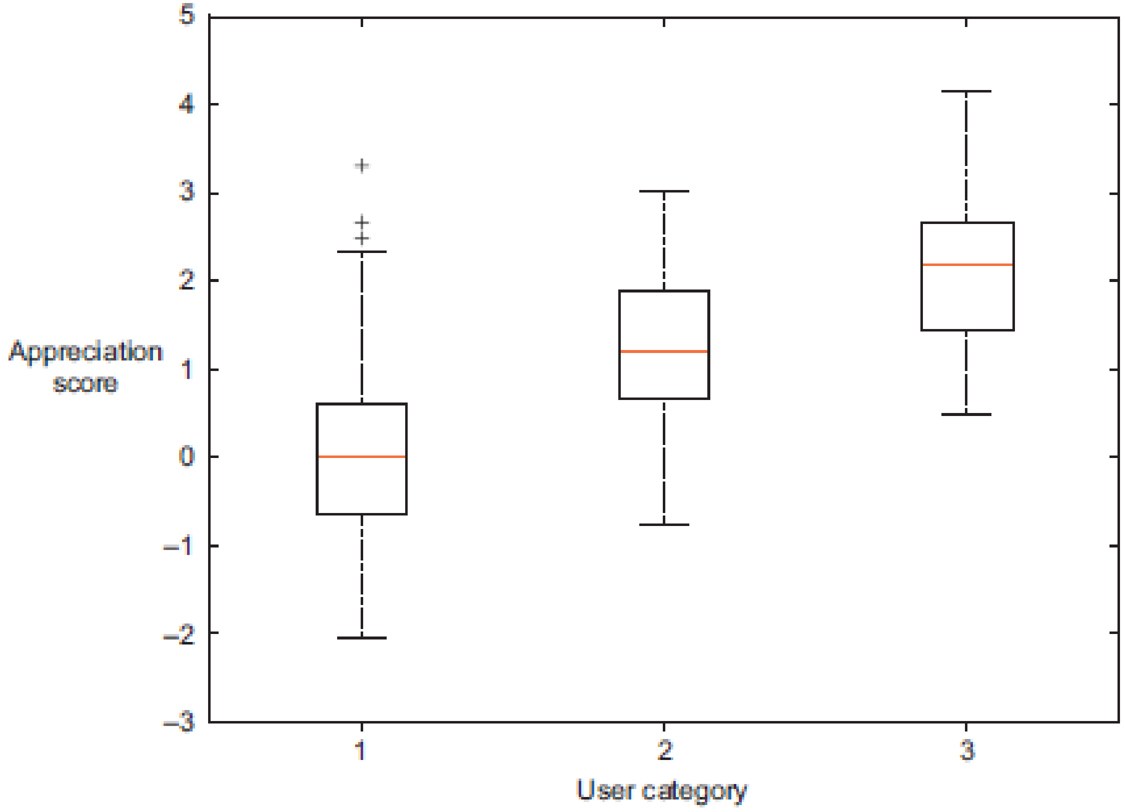
Step 4: Exploratory data analysis

- Example histogram: the number of people in the age groups of 5-year intervals

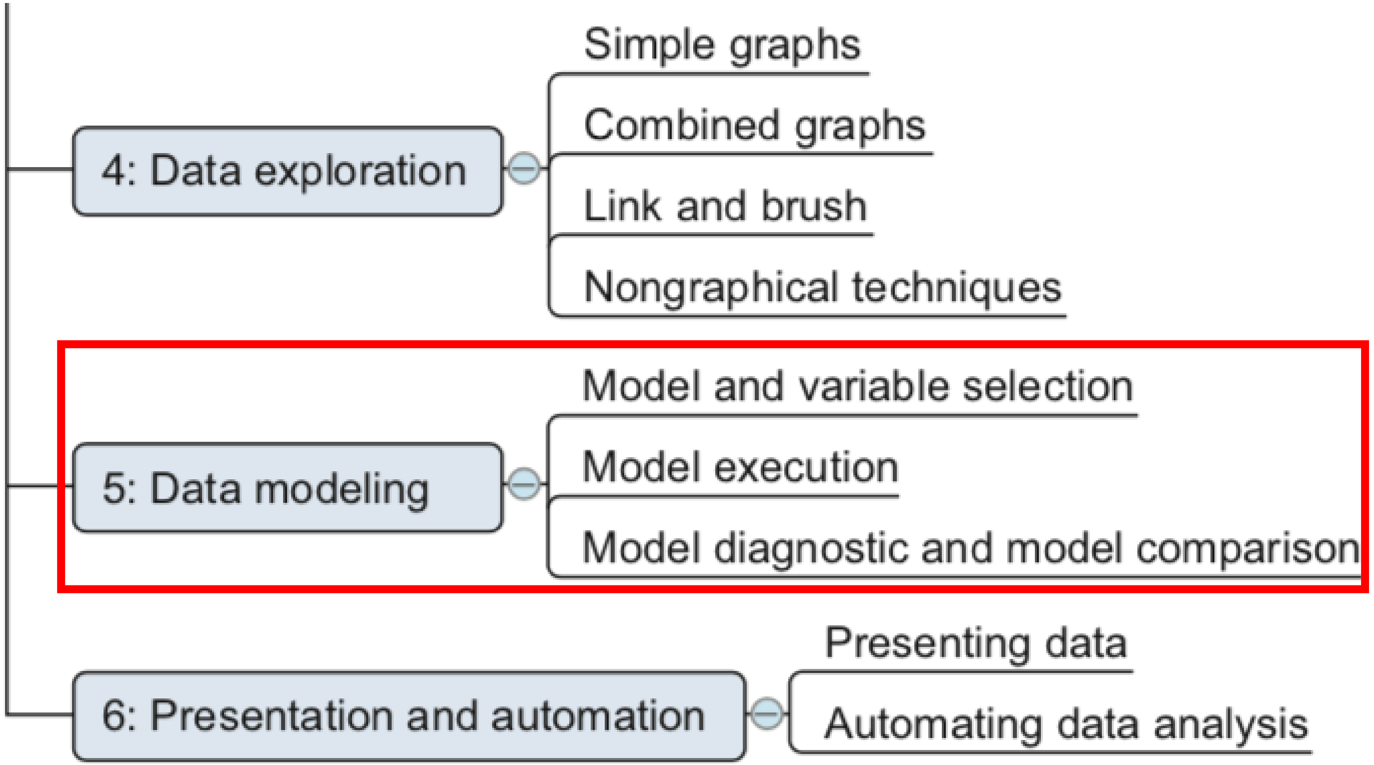


Step 4: Exploratory data analysis

- Example boxplot: each user category has a distribution of the appreciation each has for a certain picture on a photography website.



Step 5: Build the models



Step 5: Build the models

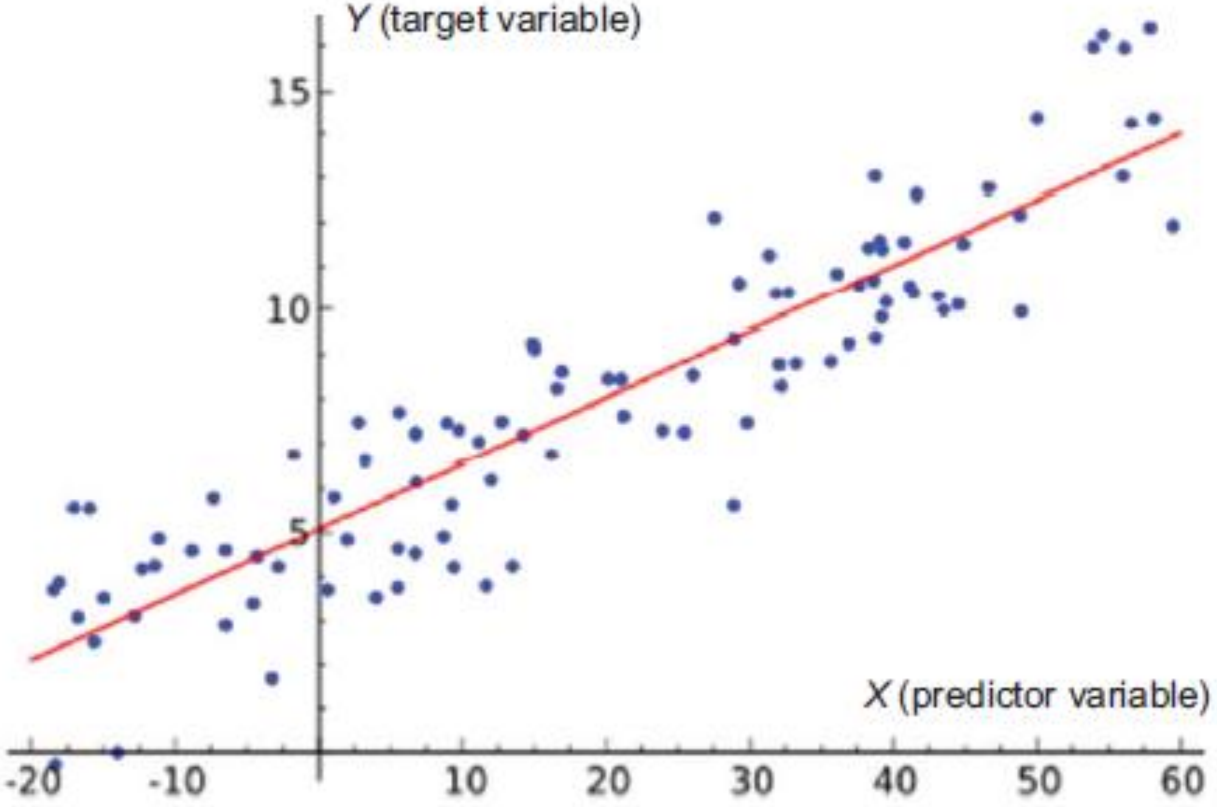
- Model and variable selection
  - Must the model be moved to a production environment and, if so, would it be easy to implement?
  - How difficult is the maintenance on the model: how long will it remain relevant if left untouched?
  - Does the model need to be easy to explain?

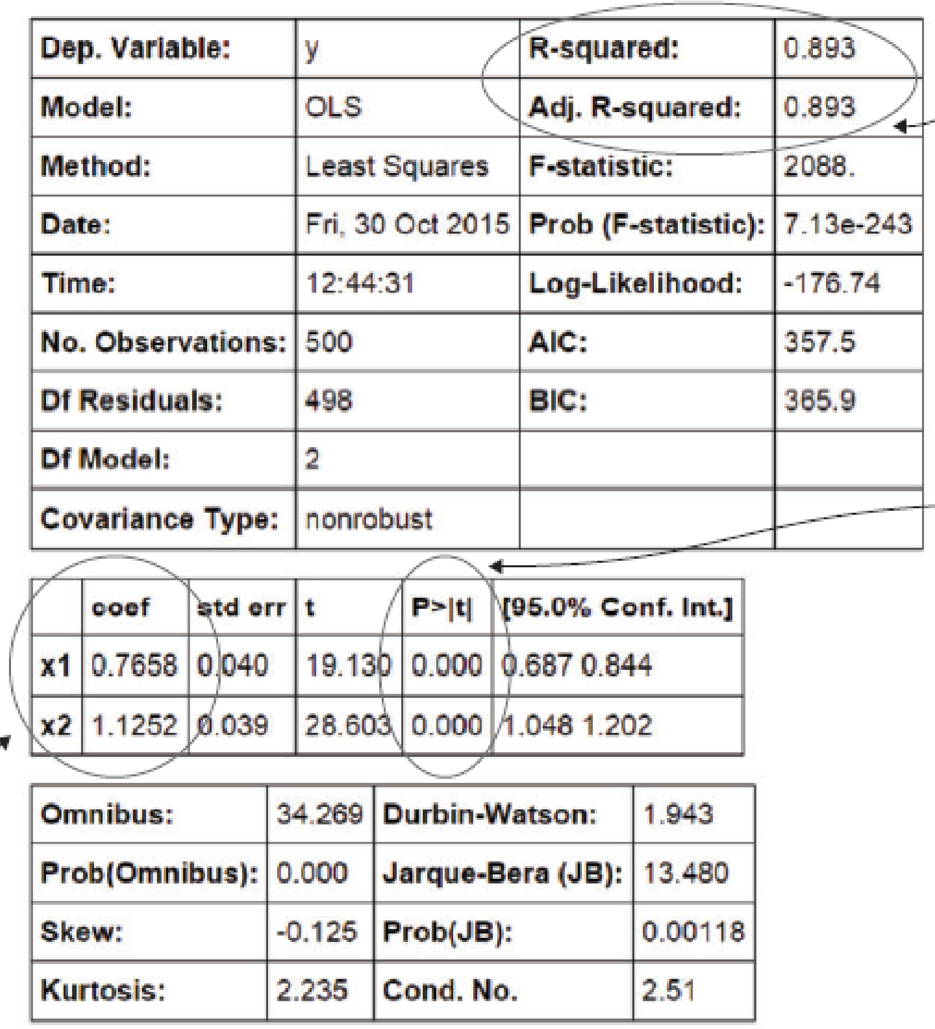
Step 5: Build the models

- Model execution
  - Once you’ve chosen a model you’ll need to implement it in code.
  - Python libraries – StatsModels or Scikit-learn

Step 5: Build the models

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





Dep. Variable:	y	R-squared:	0.893
Model:	OLS	Adj. R-squared:	0.893
Method:	Least Squares	F-statistic:	2088.
Date:	Fri, 30 Oct 2015	Prob (F-statistic):	7.13e-243
Time:	12:44:31	Log-Likelihood:	-176.74
No. Observations:	500	AIC:	357.5
Df Residuals:	498	BIC:	365.9
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
x1	0.7658	0.040	19.130	0.000	0.687 0.844
x2	1.1252	0.039	28.603	0.000	1.048 1.202

Omnibus:	34.269	Durbin-Watson:	1.943
Prob(Omnibus):	0.000	Jarque-Bera (JB):	13.480
Skew:	-0.125	Prob(JB):	0.00118
Kurtosis:	2.235	Cond. No.	2.51

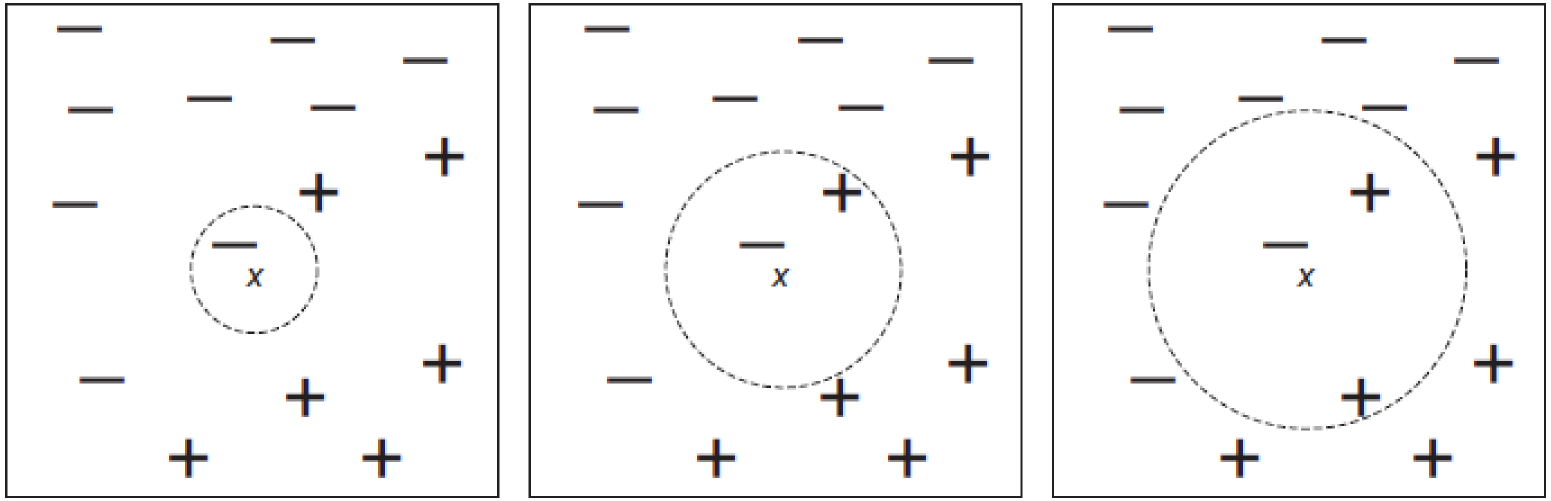
Model fit: higher is better but too high is suspicious.

p-value to show whether a predictor variable has a significant influence on the target. Lower is better and <0.05 is often considered “significant.”

Linear equation coefficients.
$y = 0.7658xl + l.l252x2$.

n

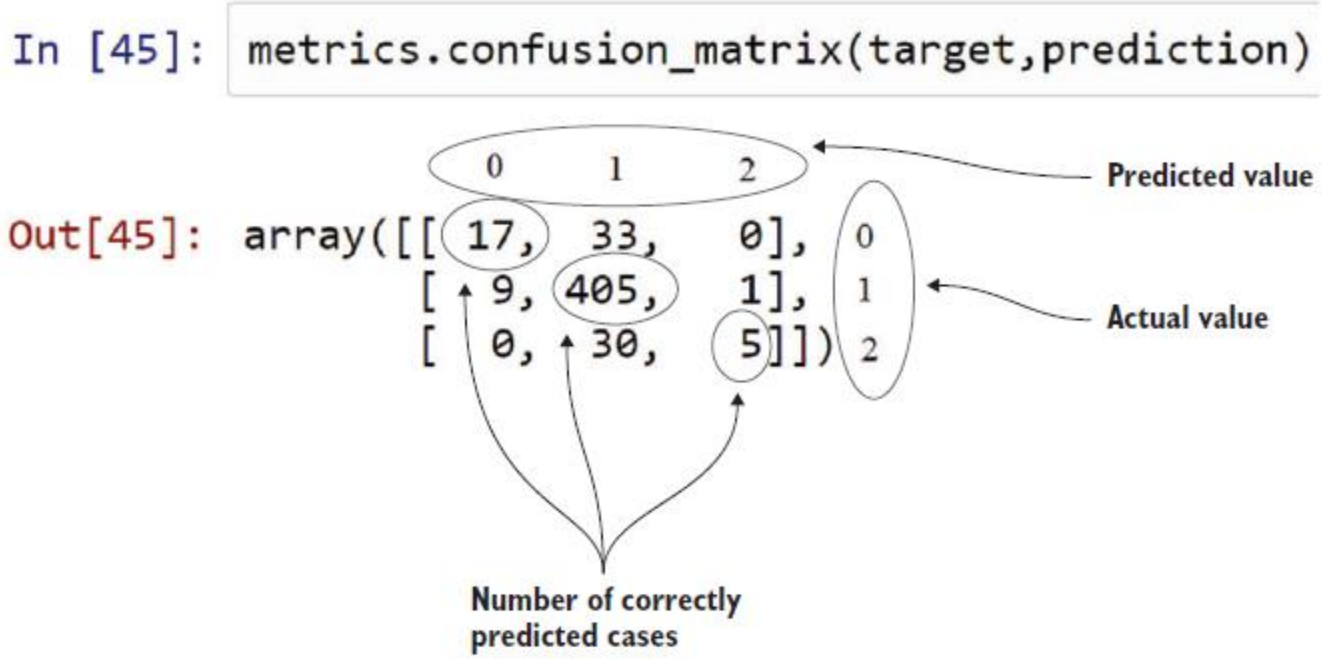
Step 5: Build the models



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

- K-nearest neighbor techniques look at the k-nearest point to make a prediction.

Step 5: Build the models



```
In [45]: metrics.confusion_matrix(target,prediction)

Out[45]: array([[ 17,  33,   0],
                [  9, 405,   1],
                [  0,  30,   5]])
```

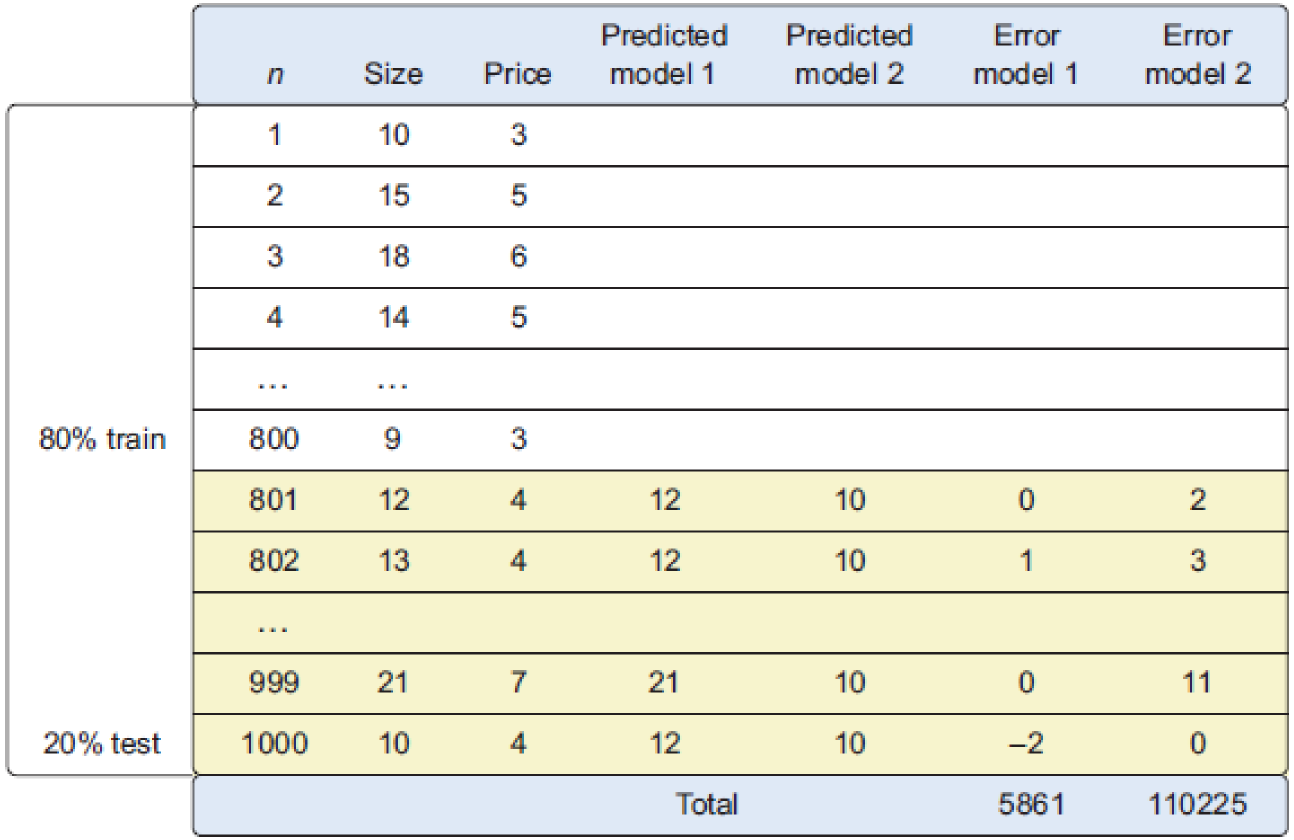
- Confusion matrix: it shows how many cases were correctly classified and incorrectly classified by comparing the prediction with the real values. Remark: the classes (0,1,2) were added in the figure for clarification.

Step 5: Build the models

- Model diagnostics and model comparison
  - 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.

Step 5: Build the models

- A holdout sample helps you compare models and ensures that you can generalize results to data that the model has not yet seen.



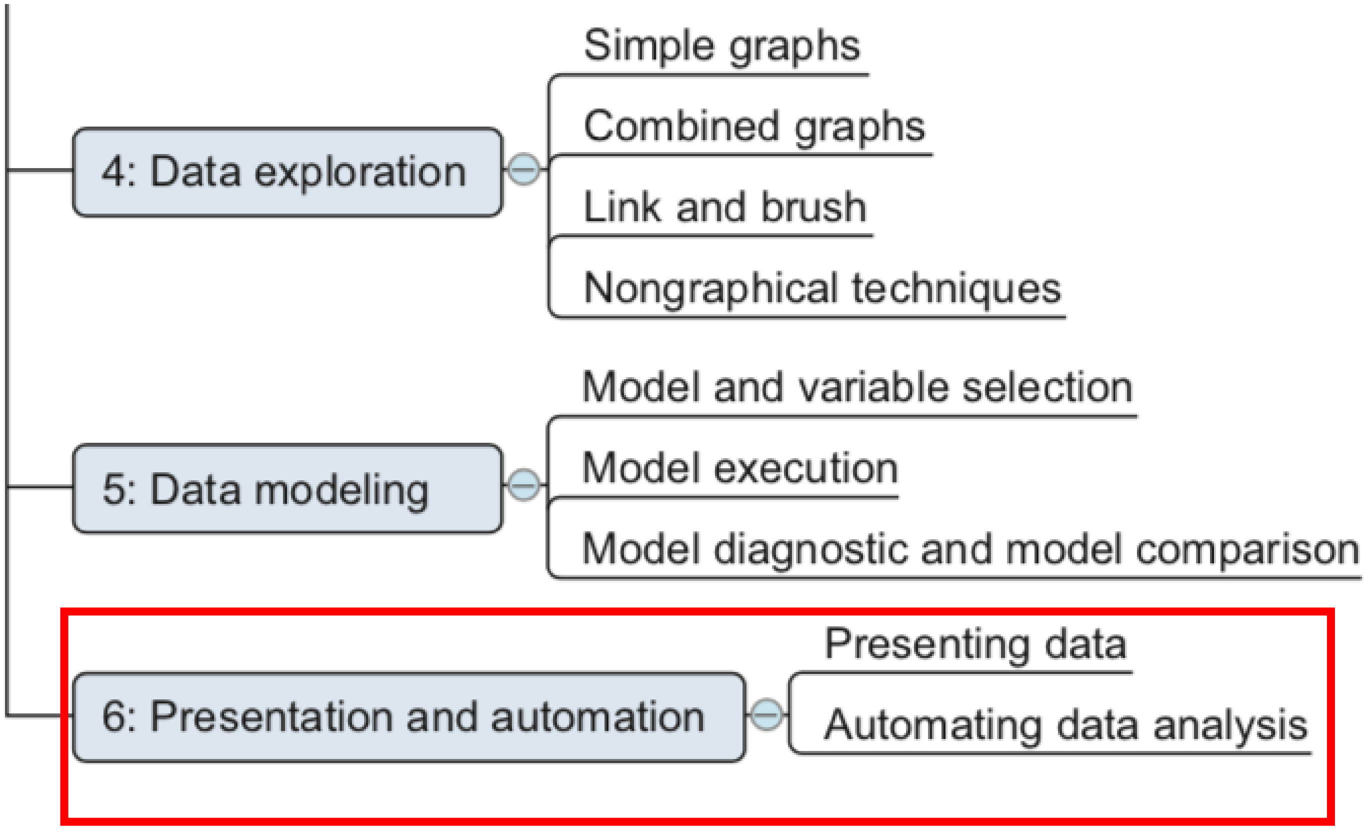
	n	Size	Price	Predicted model 1	Predicted model 2	Error model 1	Error model 2
	1	10	3				
	2	15	5				
	3	18	6				
	4	14	5				
	…	…					
80% train	800	9	3				
	801	12	4	12	10	0	2
	802	13	4	12	10	1	3
	…						
	999	21	7	21	10	0	11
20% test	1000	10	4	12	10	–2	0
				Total		5861	110225

Step 5: Build the models

- 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.

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

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



4: Data exploration

- Simple graphs
- Combined graphs
- Link and brush
- Nongraphical techniques

5: Data modeling

- Model and variable selection
- Model execution
- Model diagnostic and model comparison

6: Presentation and automation

- Presenting data
- Automating data analysis

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

- This is an exciting part; all your hours of hard work have paid off and you can explain what you found to the stakeholders.
- It is where your soft skills will be most useful, and yes, they’re extremely important.