Cleaning, tidying and preprocessing data

From data analysis to Data Science

“It is often said that 80% of data analysis is spent on the process of cleaning and preparing the data (Dasu and Johnson 2003). Data preparation is not just a first step, but must be repeated many over the course of analysis as new problems come to light or new data is collected.”

“Despite the amount of time it takes, there has been surprisingly little research on how to clean data well. Part of the challenge is the breadth of activities it encompasses: from outlier checking, to date parsing, to missing value imputation. To get a handle on the problem, this paper focusses on a small, but important, aspect of data cleaning that I call data tidying: structuring datasets to facilitate analysis.”

Extract from Tidy Data by Hadley Wickham (another important figure in Data Science)

Cleaning and preprocessing/exploration

The following useful classification is based on the forthcoming book by **Yu and Barter, “Veridical Data Science”**, from which we take a few examples later in this lecture

- Cleaning: fixing redeemable issues with the data
- Preprocessing/exploration: representing the data so that it can be conveniently and successfully (e.g., better predictability) used by algorithms in downstream analysis

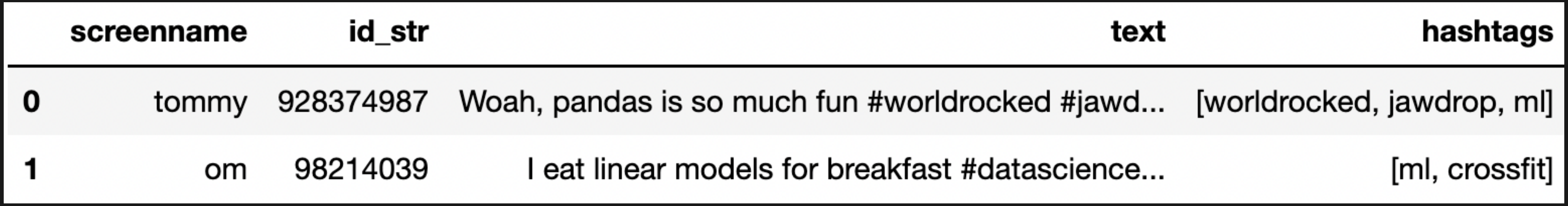
- We will understand these two concepts, mainly by a number of examples and counterexamples!
- We will highlight good practices and important considerations
- Eventually, Data Science aspires to adopting a **scientific** approach to all aspects of working with Data - as opposed to irreproducible hacks and mysterious manipulations
- The following is a scientific framework for **tidy data**

Tidy data

Context

- The assumption in this part is that we are given a dataset that contains information on variables and individuals
- Later in this lecture we will consider how to represent numerically categorical data, and how to deal with the important and common issue of missing data

- The aim in this part is to expose you to some good practices for how to preprocess, and in effect, tidy the dataset so that it then can be used within statistical learning algorithms
- This is not a panacea, neither the way to go in each situation. However, it is one *scientific* approach to work with datasets
- A good example of an *untidy* dataset is that of the tweets that we analyzed in the pandas notebook

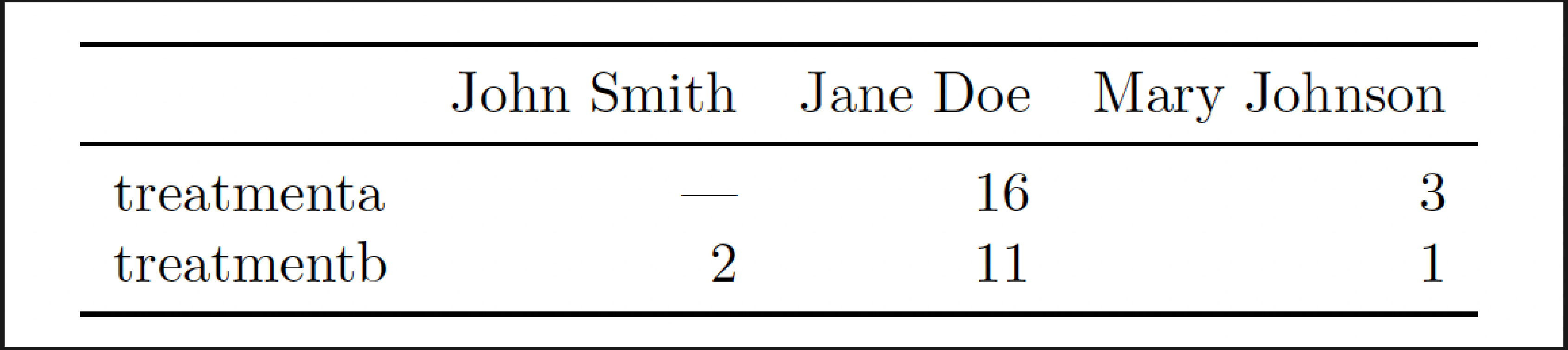


	screenname	id_str	text	hashtags
0	tommy	928374987	Woah, pandas is so much fun #worldrocked #jawd...	[worldrocked, jawdrop, ml]
1	om	98214039	I eat linear models for breakfast #datascience...	[ml, crossfit]

Data semantics

- Values
- Variables
- Observations/examples/units etc

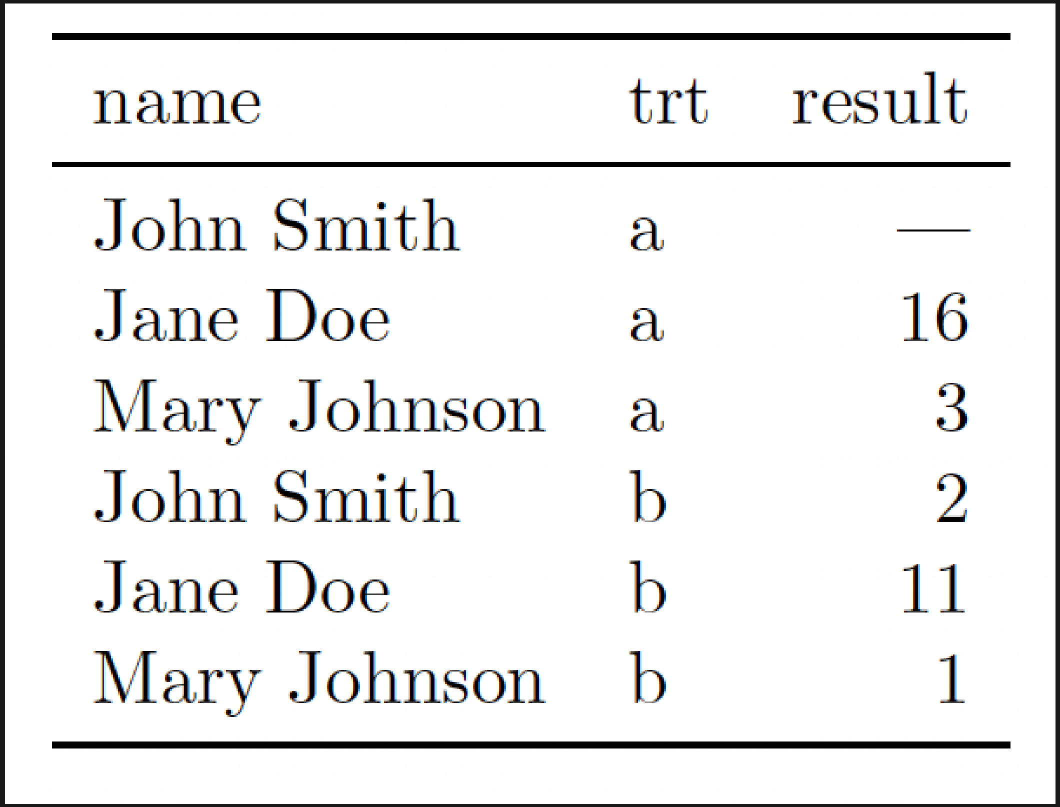
Every value belongs to a variable and an observation. In the following example the organization is compact but *messy*



	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

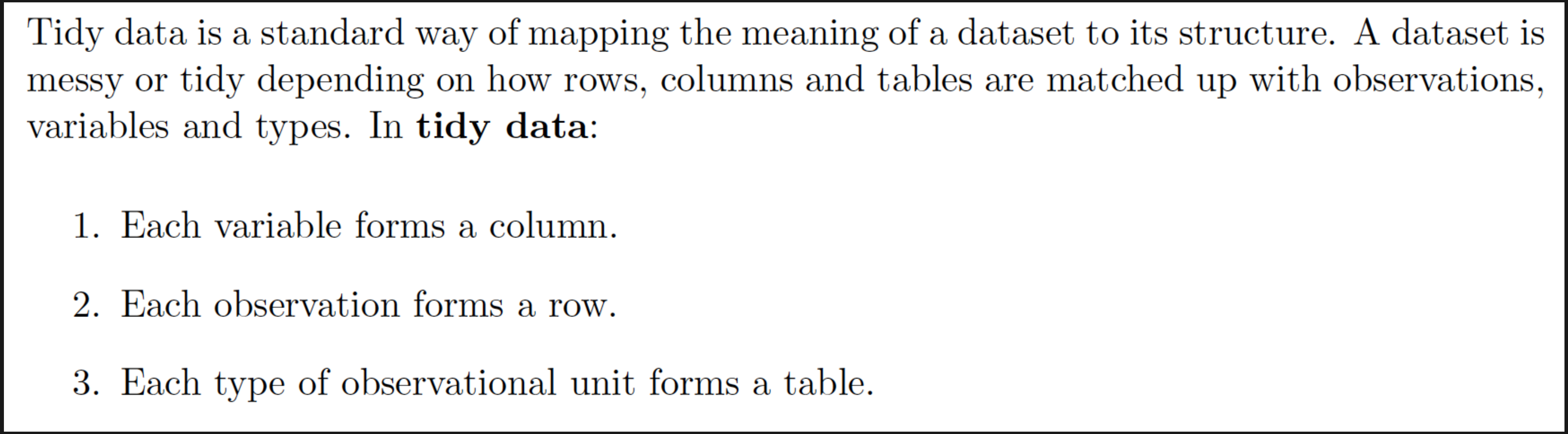
From Tidy Data

Here is a different, *tidy* organization of the same values: each row is an observation, each column is a variable



name	trt	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

From Tidy Data



Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types. In **tidy data**:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

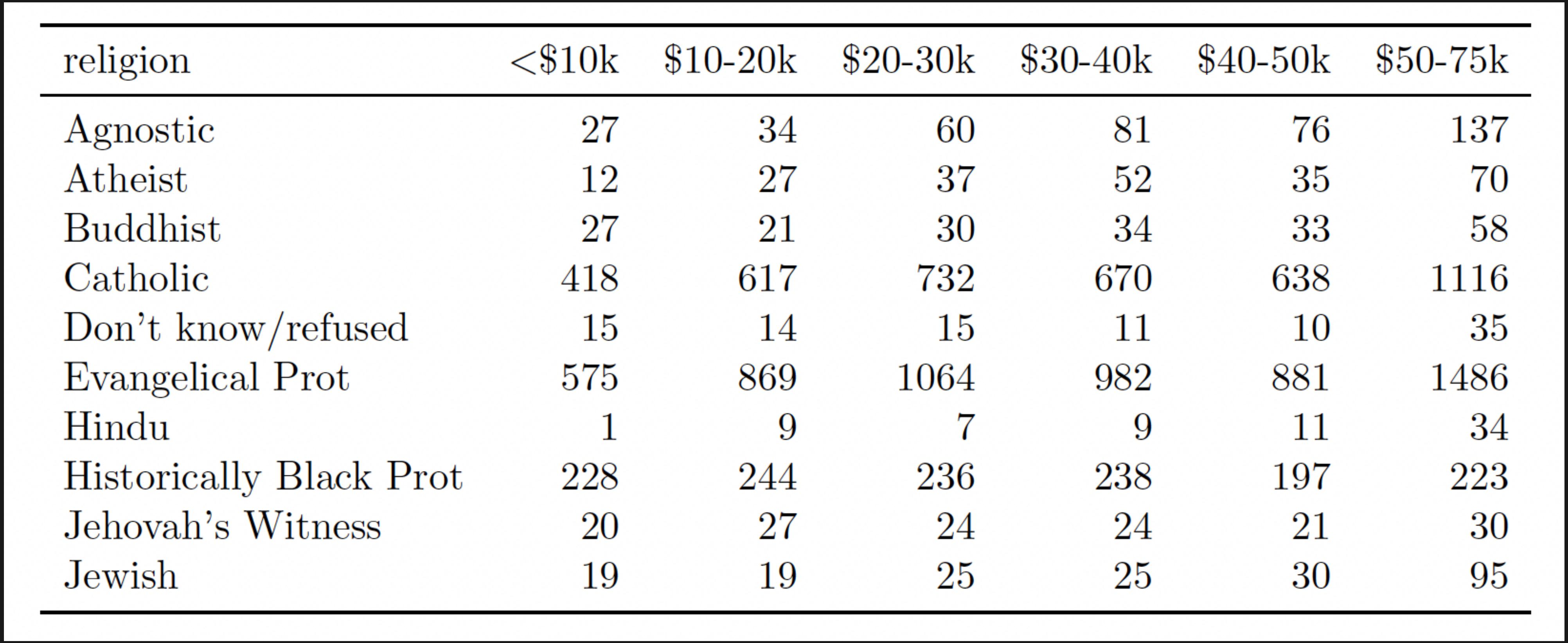
From Tidy Data

Messy data is any other other arrangement of the data

Case for tidy data

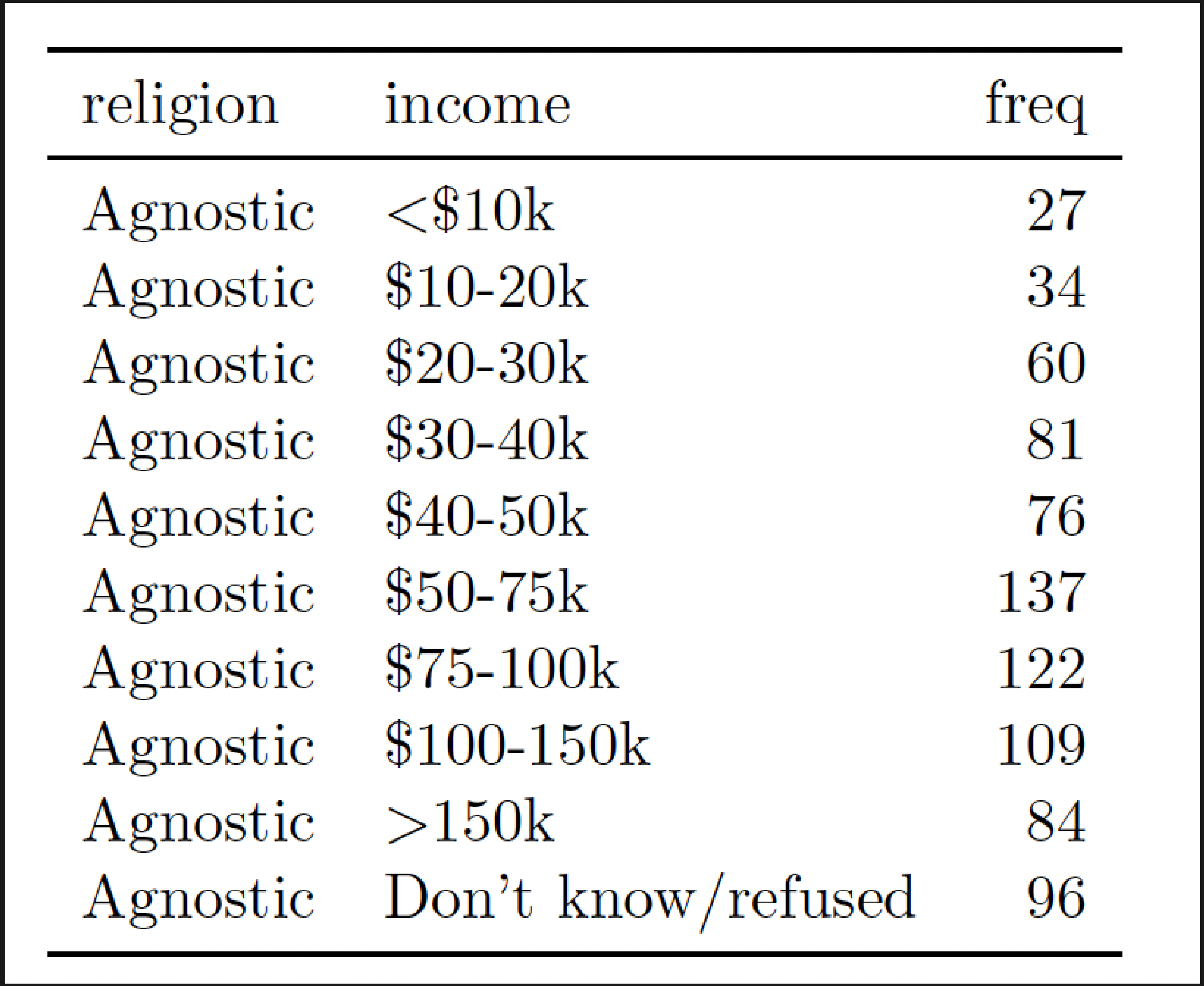
- Simple for downstream analytics
- In the example: in the messy dataset you need to use different strategies to extract different variables (e.g, plot results with colour by treatment)
- Suited to vectorised programming, such as `R` and `Python`

Example of untidy datasets: values as columns



religion	<$10k	$10-20k	$20-30k	$30-40k	$40-50k	$50-75k
Agnostic	27	34	60	81	76	137
Atheist	12	27	37	52	35	70
Buddhist	27	21	30	34	33	58
Catholic	418	617	732	670	638	1116
Don't know/refused	15	14	15	11	10	35
Evangelical Prot	575	869	1064	982	881	1486
Hindu	1	9	7	9	11	34
Historically Black Prot	228	244	236	238	197	223
Jehovah's Witness	20	27	24	24	21	30
Jewish	19	19	25	25	30	95

From Tidy Data



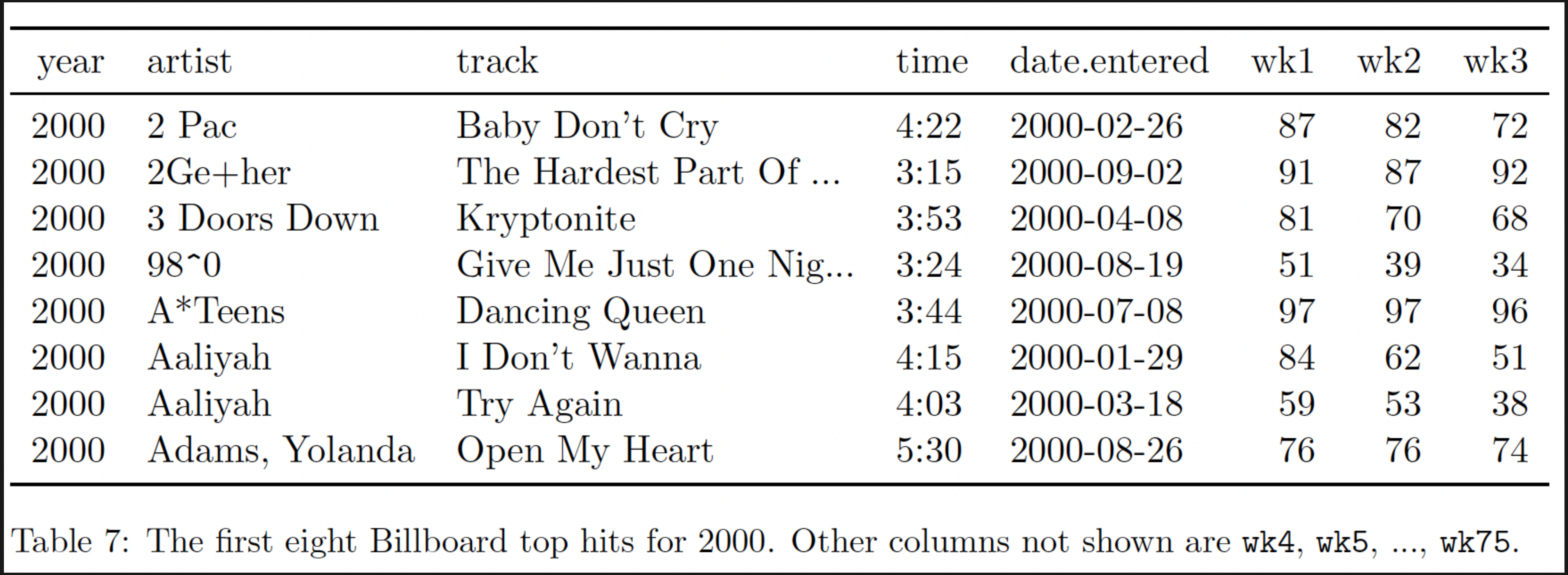
religion	income	freq
Agnostic	<$10k	27
Agnostic	$10-20k	34
Agnostic	$20-30k	60
Agnostic	$30-40k	81
Agnostic	$40-50k	76
Agnostic	$50-75k	137
Agnostic	$75-100k	122
Agnostic	$100-150k	109
Agnostic	>150k	84
Agnostic	Don't know/refused	96

From Tidy Data

This is a “long” format for the previous “wide”. In Wyckham’s terminology, a “molten” dataset

Example of untidy datasets: multiple types in one table

Consider the following compact and useful for many things, but *untidy* for further analysis, dataset!

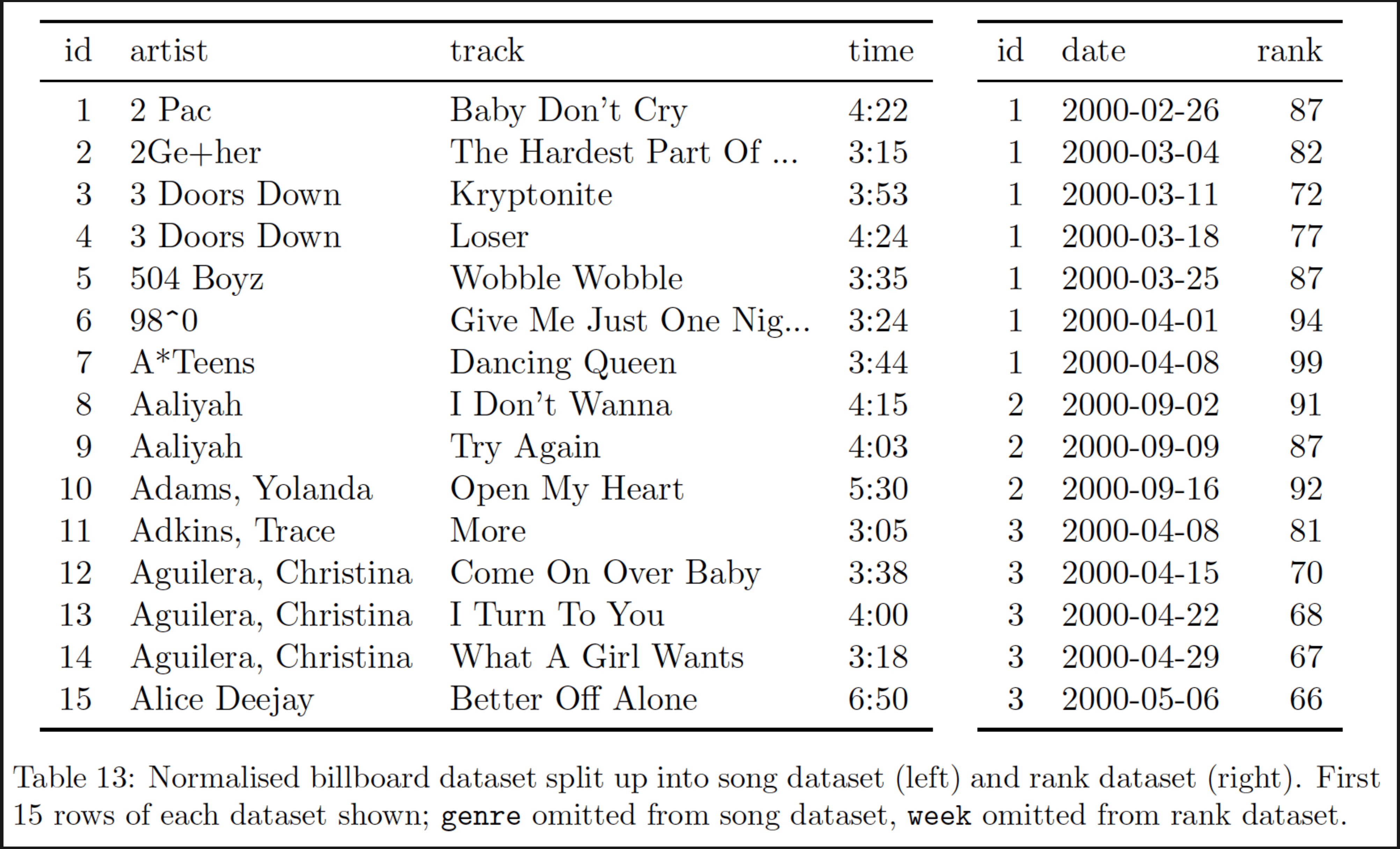


year	artist	track	time	date.entered	wk1	wk2	wk3
2000	2 Pac	Baby Don't Cry	4:22	2000-02-26	87	82	72
2000	2Ge+her	The Hardest Part Of ...	3:15	2000-09-02	91	87	92
2000	3 Doors Down	Kryptonite	3:53	2000-04-08	81	70	68
2000	98^0	Give Me Just One Nig...	3:24	2000-08-19	51	39	34
2000	A*Teens	Dancing Queen	3:44	2000-07-08	97	97	96
2000	Aaliyah	I Don't Wanna	4:15	2000-01-29	84	62	51
2000	Aaliyah	Try Again	4:03	2000-03-18	59	53	38
2000	Adams, Yolanda	Open My Heart	5:30	2000-08-26	76	76	74

Table 7: The first eight Billboard top hits for 2000. Other columns not shown are `wk4`, `wk5`, ..., `wk75`.

From Tidy Data

- One issue is the “values as columns” problem
- Another is that there are really 2 types of information conflated. There are song-metadata and position billboard data. A tidy dataset should be in two tables that can be merged later if needed (**recall the “customer of the month” problem**)



id	artist	track	time
1	2 Pac	Baby Don't Cry	4:22
2	2Ge+her	The Hardest Part Of ...	3:15
3	3 Doors Down	Kryptonite	3:53
4	3 Doors Down	Loser	4:24
5	504 Boyz	Wobble Wobble	3:35
6	98^0	Give Me Just One Nig...	3:24
7	A*Teens	Dancing Queen	3:44
8	Aaliyah	I Don't Wanna	4:15
9	Aaliyah	Try Again	4:03
10	Adams, Yolanda	Open My Heart	5:30
11	Adkins, Trace	More	3:05
12	Aguilera, Christina	Come On Over Baby	3:38
13	Aguilera, Christina	I Turn To You	4:00
14	Aguilera, Christina	What A Girl Wants	3:18
15	Alice Deejay	Better Off Alone	6:50

id	date	rank
1	2000-02-26	87
1	2000-03-04	82
1	2000-03-11	72
1	2000-03-18	77
1	2000-03-25	87
1	2000-04-01	94
1	2000-04-08	99
2	2000-09-02	91
2	2000-09-09	87
2	2000-09-16	92
3	2000-04-08	81
3	2000-04-15	70
3	2000-04-22	68
3	2000-04-29	67
3	2000-05-06	66

Table 13: Normalised billboard dataset split up into song dataset (left) and rank dataset (right). First 15 rows of each dataset shown; `genre` omitted from song dataset, `week` omitted from rank dataset.

From Tidy Data

Data cleaning

Agenda

- Cleaning: fixing redeemable issues with the data
- Some redeemable issues:
  - variable types
  - column names
  - dimensions
  - etc etc etc

But let's take a systematic approach guided through some examples. We will see parts of an interesting case study that is carried out in the forthcoming book by Yu and Barter, "Veridical Data Science"

They consider a publicaly available dataset on organ donations with view to understanding the time-trends of organ donation and how these vary across the world. The data is available here:

https://www.transplant-observatory.org/export-database/

Actually, in principle the life cycle of a Data Science project should start with more refined questions.

The way it is described above is more *exploratory*. For our purposes this is more or less OK since it is used to highlight some data cleaning issues. Although, a good understanding of what one expects to learn from this data does interact even with some cleaning decisions

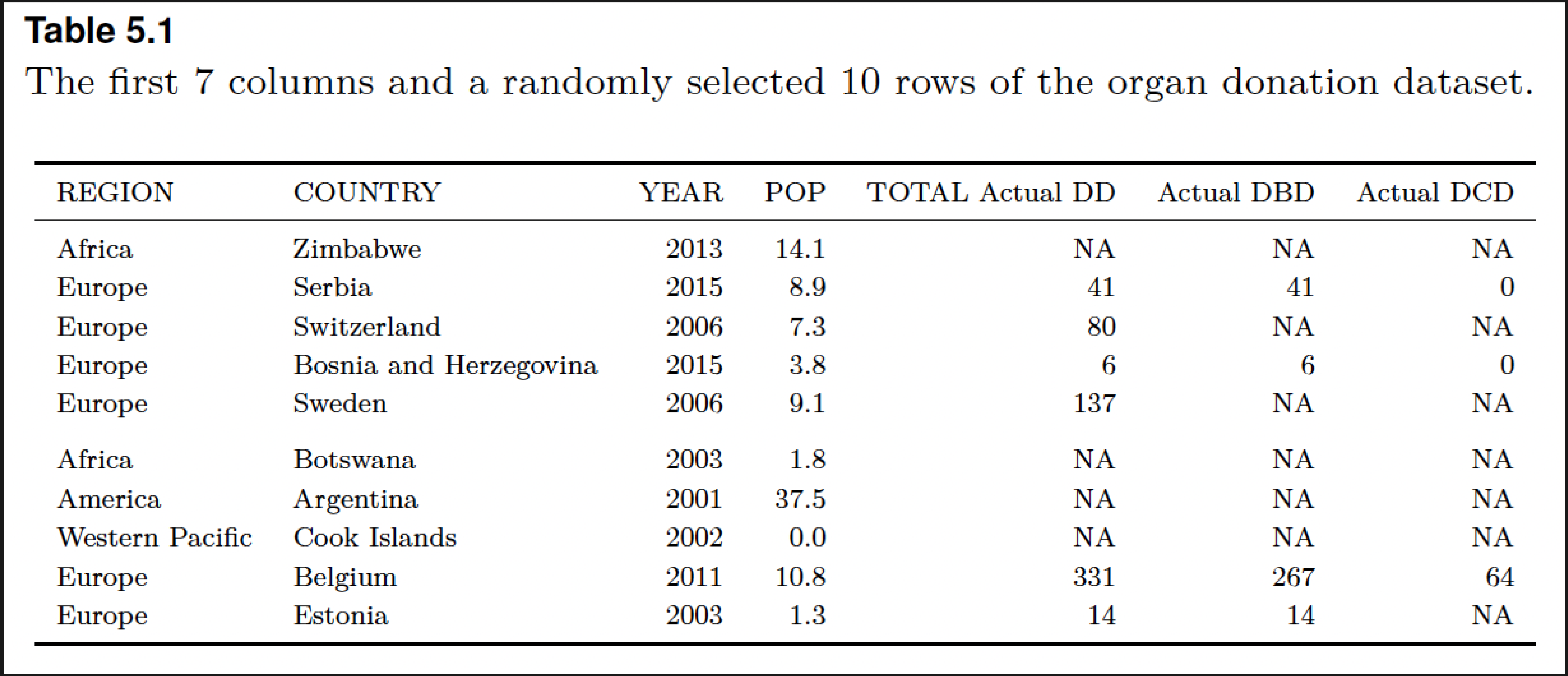


Table 5.1

The first 7 columns and a randomly selected 10 rows of the organ donation dataset.

REGION	COUNTRY	YEAR	POP	TOTAL Actual DD	Actual DBD	Actual DCD
Africa	Zimbabwe	2013	14.1	NA	NA	NA
Europe	Serbia	2015	8.9	41	41	0
Europe	Switzerland	2006	7.3	80	NA	NA
Europe	Bosnia and Herzegovina	2015	3.8	6	6	0
Europe	Sweden	2006	9.1	137	NA	NA
Africa	Botswana	2003	1.8	NA	NA	NA
America	Argentina	2001	37.5	NA	NA	NA
Western Pacific	Cook Islands	2002	0.0	NA	NA	NA
Europe	Belgium	2011	10.8	331	267	64
Europe	Estonia	2003	1.3	14	14	NA

From Veridical Data Science

The first, most fundamental and often omitted step!

- find out how data was collected
- if part of a study what purposes and if there are previous analyses
- if there are proprietary and privacy issues
- identify additional/alternative sources of data that could be useful
- collect the above in a project documentation

A first checklist

- understand what each variable measures
- check type and if correctly read by computer
- structural missingness & decisions
- valid entries (within range from domain knowledge)
- standard notation for missing (NA, NaN, etc, but not 99 or 0)
- column names

- check if structural constraints are met (e.g if a variable is meant to be the sum of others)
- dimension check
- variable types (especially time/date) & common type per column
- sensible coding of categorical variables
- units of measurements (especially if there are inconsistencies between columns, e.g. one measures in units and another in 1000ds of units)
- calls on omitting variables with high % of missingness

Missing data

Warning

This is a **massive** research area of great importance to applied Data Science. It is not just a matter of doing a couple of tricks. It is also closely related to **causal inference**, which can be formulated as a missing data problem

The purpose here is to illustrate the issues and provide some first guidance

Structural missingness

This refers to entries in the dataset that are *un-measurable*, i.e., their values would not make sense

For example, the answer to a question of whether a person has been pregnant would make no sense for a male respondent, therefore, the corresponding entry in a table would be empty

This is an example of structural missingness

Here’s a question for you to think about: in a dataset that consists of male and female respondents, where one of the variables is the sex of the respondent and another the answer to whether they have been pregnant, how could you transform the dataset with the empty cells to one that is tidy?

Non-structural missingness

This is pervasive in datasets even when collected via carefully designed surveys or experiments. Some entries in the dataset are missing, but there is a well-defined value that should have been measured but for different reasons it has not been

It is useful to return to the organ donation dataset and think about missingness there

Main concepts for non-structural missingness

The following can be defined mathematically but we will stick to the high-level concepts

- Missing completely at random (MCAR): an entry is missing independently of the information in the dataset
    - e.g., the person doing the data entry erased by mistake some entries

- Missing at random (MAR): the missingness of an entry is predictable from the non-missing variables
  - e.g, the probability not to respond to the pregnancy question depends on the socio-demographic stratum that the woman belongs to and can be recovered from the information in the survey (such as ethnic and religious background) but not on the pregnancy statusa

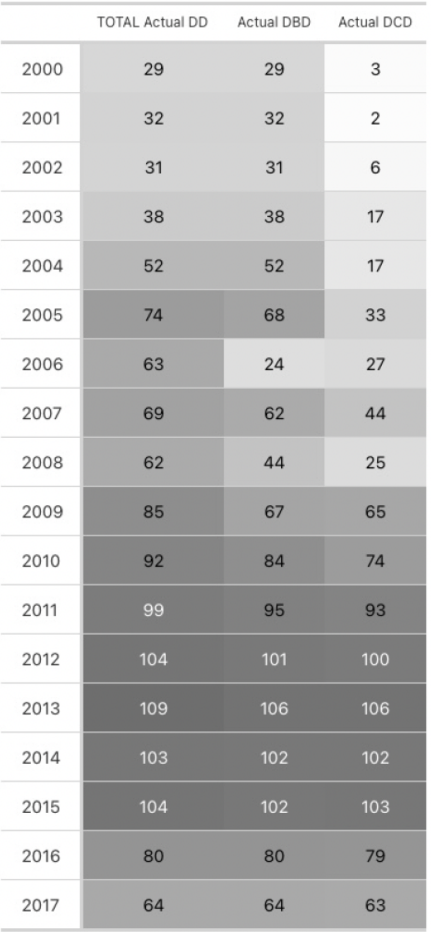
- Missing not at random (NMAR): everything less, effectively the probability that an entry is missing depends on the unobserved value
  - e.g., women with terminated pregnancy are more likely not to respond
  - e.g., 0 population in a dataset that reports country population in units of millions

This example is from Little and Rubin’s book on Missing Data, which is an extensive resource on the topic:

“Suppose Y1 is age, and Y2 is income. If the probability that income is missing is the same for all individuals, regardless of their age or income, then the data are MCAR. If the probability that income is missing varies according to the age of the respondent but does not vary according to the income of respondents with the same age, then the data are MAR. If the probability that income is recorded varies according to income for those with the same age, then the data are NMAR.”

- These are assumptions, *non-testable* from the dataset itself
- It is particularly useful to explore the the pattern of missingness, this can often - together with subject matter knowledge - provide explainations and even suggest the type of missingness

For example, returning on the organ donation data:



	TOTAL Actual DD	Actual DBD	Actual DCD
2000	29	29	3
2001	32	32	2
2002	31	31	6
2003	38	38	17
2004	52	52	17
2005	74	68	33
2006	63	24	27
2007	69	62	44
2008	62	44	25
2009	85	67	65
2010	92	84	74
2011	99	95	93
2012	104	101	100
2013	109	106	106
2014	103	102	102
2015	104	102	103
2016	80	80	79
2017	64	64	63

Figure 5.1
The number of countries reporting non-missing counts by year for three of the numeric variables. The darker the color of the cell, the more countries reported data for the corresponding variable (column) in that year (row).

From Veridical Data Science

Methodology for missing data

- Course of action depends on the type of missingness (MCAR etc) and the *pattern* of missingness (when not MCAR)
- Imputation methods are based on **predictive algorithms** to fill in missing values, typically in a MAR framework
  - Advisable to create *several* imputed datasets
  - Sophistication varies from mean and nearest value imputation, to regression imputation or even *generative models*

- Joint analysis methods built a joint for model for observed and missing data
  - This capitalises on computational methods for latent variable models, including **EM algorithm, MCMC, variational inference**
  - matrix factorization models and algorithms are a good example in this framework
  - Imputation methods are a computationally cheap (and statistically less appropriate) version of this. They can also be based on this, e.g. using matrix factorization.
  - Clearly, this is entirely within the *modelling* stage

- Complete case analysis drops all rows with missing entries in some comlumns. This has two major problems:
    - For MCAR leads to valid but *statistically ineffecient* inference
        - Say that each cell is MCAR with prob. 0.05. Hence 5% of the entries in the dataset are missing. Then, even for a dataset with 20 variables, if we drop each row with a missing entry we would be droppping on average 65% of the rows!! (you might want to think how we arrived to this number..)
    - For MAR or MNAR this leads to selection bias

Preprocessing

Data splits

- A convincing approach to show **predictability** of a model/algorithm is to show it’s capacity to predict data that have not been used in the training but are from *the context of interest*
  - A very strict but precise way to make concrete the italicized expression: unseen data distributed as the training data
- This is useful in our analysis and post-analysis but “pre-commercialization” stage

- Often new data are not easy to find so we instead keep aside part of the data before any analysis is done, and use it later for assessing predictability; this is called the test dataset
- It turns out, as we will see, that it is useful to also keep aside another part for validation of tuning hyperparameters; this is called the validation dataset
- The remaining data called the training dataset will be used for training

- It is not feasible to come up with a scientific framework for how to do the split; e.g., common ad-hoc practice is to keep 20% for test
- Later we will understand better the pros and cons of making each of these datasets smaller or larger. You can already understand that you would want all three to be as large as possible but this is infeasible!
- These splits are closely related to the concept of cross-validation, which we will understand in future lectures and for which some rigorous answers are available

- A similarly tricky problem is which data to include in the splits. The ideal is that the data in each part are a good reflection of the data we wish to use our models for
- A trivial, common but often inappropriate way to split is to randomly assign each available observation to one of the parts
- When data are structured, such as organised in space or time, further care is needed

- For example, the housing dataset that you will use in your project consists of real estate prices recorded over several years, where the recorded input variables are house characteristics and the time of the sale, and the output variable is the sale price
- One might consider training a model using the first so-many years, and keep the last so-many years as test.

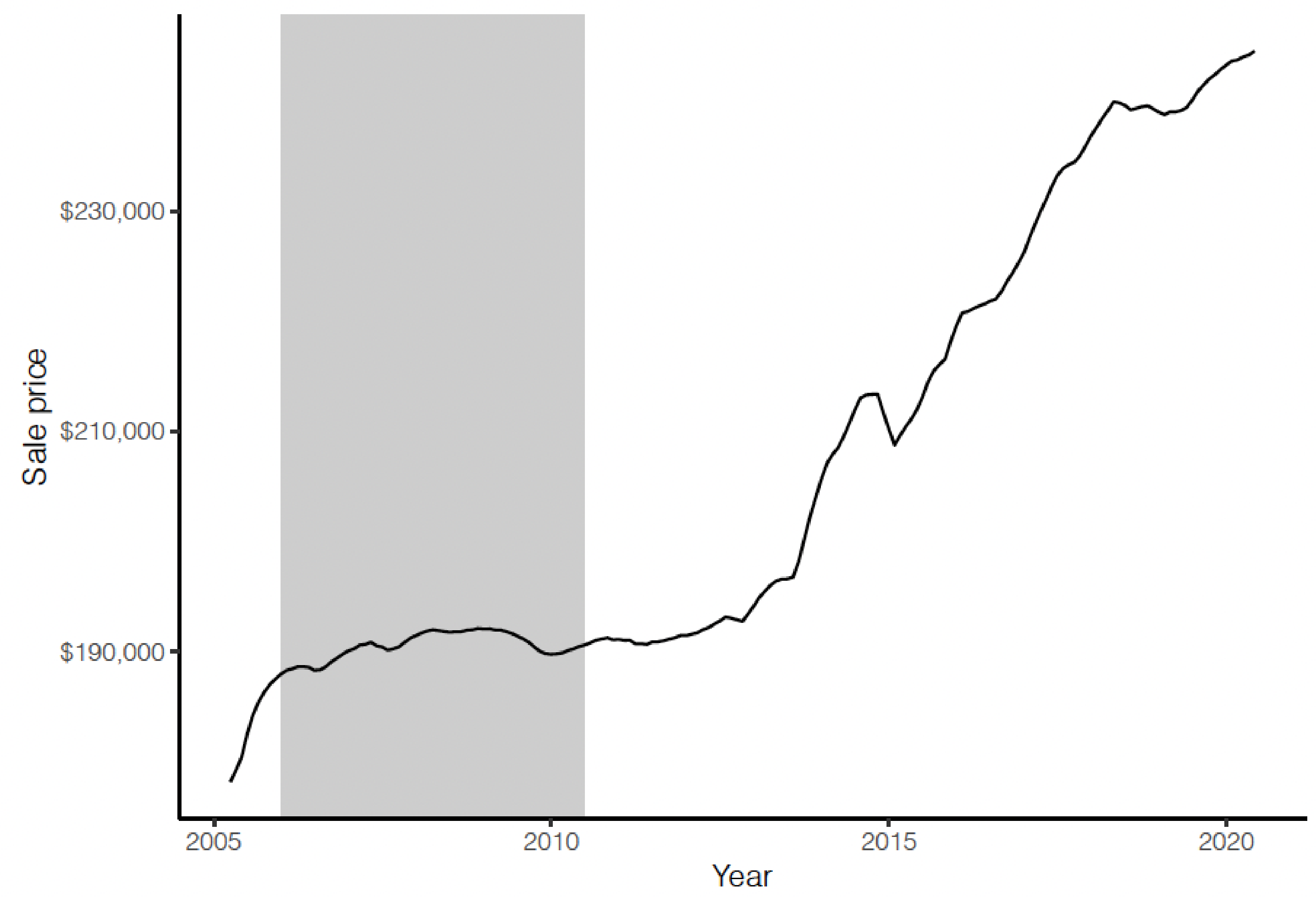


Figure 9.2
House price trends in Ames, Iowa based on data from Zillow. The time period covered in De Cock's data is highlighted using a grey rectangle.

From Veridical Data Science

Common practices

- Highly correlated variables
  - Various algorithms might face (statistical and computational) problems if dataset contains very highly correlated inputs
  - Using also subject matter knowledge we might wish to drop some from the dataset before analysis
    - Typically OK for prediction
    - Can be catastrophic for causal inference!

- Transformations
    - For various algorithms it is important that input variables have been scaled so that they have a comparable rage
        - E.g., sample mean 0 and sample var 1
        - We will see later an example of massive instability to this in the context of neural nets.
        - Lasso and other regularisation methods require this - we will understand why

- Representation of categorical variables
    - This is highly dependent on the software used for the analysis
    - A common approach is to use a 1-hot-encoding that uses *dummy variables*

- “Outliers” pt 1
  - This is a major misconception and an area (as with missing data) where often practitioners do very silly things and where Data Science from hacking differs a lot
  - Practitioners often replace *large* values of a variable by a more central value such as the median
  - The short story here is that there is no short story!

- “Outliers” pt 1
  - Key concepts that relate to this theme is: robustness, leverage, influence
  - It is:
    - **good** to be aware of observations with high-values in some variables
    - **necessary** to explore by box-plots all variables
    - **advisable** not to change the data but deal with the issue using stability concepts instead - or even models that account for heterogeneity

The good Data Scientist

Main principles

- Document your notes about the origin and purpose of the dataset
- Document your calls on cleaning and preprocessing
- **NEVER** change the original dataset using Excel. Instead, write a program that performs all the required steps, it takes as input the original dataset and returns one or more clean and preprocessed dataset(s) and it is compatible with what is described in the documentation