



24 JANUARY 2019

Herman Teirlinck,
01.71 - Frans Breziers

What have I done?!?

<https://ropensci.org/blog/2019/01/22/waterinfo-tidal-eel/>

The image shows a screenshot of a web browser displaying a Twitter post. The browser's address bar shows the URL <https://twitter.com/rOpenSci/status/1087724418288123905>. The Twitter post is from the account **rOpenSci** (@rOpenSci). The text of the tweet reads: "[blog] "waterRinfo - Downloading tidal data to understand the behaviour of a migrating eel" by @SVanHoey & @peterdesmet of @LifeWatchINBO". Below the text is a link to the blog: [ropensci.org/blog/2019/01/2 ...](https://ropensci.org/blog/2019/01/22/waterinfo-tidal-eel/) and the hashtag **#rstats**. The tweet includes two images: on the left, a stylized illustration of green and blue vertical shapes resembling eels or water columns; on the right, a map showing a coastal area with a network of lines and several red dots indicating specific locations. The background of the browser shows the Twitter interface with navigation links like "Home" and "About", and a sidebar with "New to Twitter" and "You may also" sections.

The logo consists of a light green rounded rectangle with a white border. Inside the rectangle, the words "TIDY" and "DATA" are written in white, uppercase, sans-serif font, stacked vertically and centered.

TIDY
DATA

What does tidy data mean?

The image shows a Google Translate interface. At the top, there are language selection buttons: "DETECT LANGUAGE", "DUTCH", "ENGLISH" (highlighted with a blue underline), and "SPANISH" with a dropdown arrow. In the center, there is a bidirectional arrow icon and another set of language selection buttons: "DUTCH" (highlighted with a blue underline), "ITALIAN", and "ENGLISH" with a dropdown arrow. The main content area is split into two panels. The left panel, representing the source language (English), contains the word "tidy" in a large font, a close button (X), the phonetic transcription "ˈtɪdē", a microphone icon, a speaker icon, and a character count "4/5000" with a keyboard icon. The right panel, representing the target language (Dutch), contains the translation "ordelijk" in a large font, a star icon, a microphone icon, and icons for copy, edit, and share.



Journal of Statistical Software

August 2014, Volume 59, Issue 10.

<http://www.jstatsoft.org/>

Tidy Data

Hadley Wickham
RStudio

Abstract

A huge amount of effort is spent cleaning data to get it ready for analysis, but there has been little research on how to make data cleaning as easy and effective as possible. This paper tackles a small, but important, component of data cleaning: data tidying. Tidy datasets are easy to manipulate, model and visualize, and have a specific structure: each variable is a column, each observation is a row, and each type of observational unit is a table. This framework makes it easy to tidy messy datasets because only a small set of tools are needed to deal with a wide range of un-tidy datasets. This structure also makes it easier to develop tidy tools for data analysis, tools that both input and output tidy datasets. The advantages of a consistent data structure and matching tools are demonstrated with a case study free from mundane data manipulation chores.

Keywords: data cleaning, data tidying, relational databases, R.

<https://doi.org/10.18637/jss.v059.i10>

un-tidy

WWTP	Treatment A	Treatment B
Destelbergen	8.	6.3
Landegem	7.5	5.2
Dendermonde	8.3	6.2
Eeklo	6.5	7.2

un-tidy

WWTP	Treatment A	Treatment B
Destelbergen	8.	6.3
Landegem	7.5	5.2
Dendermonde	8.3	6.2
Eeklo	6.5	7.2

tidy

WWTP	Treatment	pH
Destelbergen	A	8.
Landegem	A	7.5
Dendermonde	A	8.3
Eeklo	A	6.5
Destelbergen	B	6.3
Landegem	B	5.2
Dendermonde	B	6.2
Eeklo	B	7.2

tidy data: principles

- Each **observation** forms a **row**

WWTP	Treatment	pH
Destelbergen	A	8.
Landegem	A	7.5
Dendermonde	A	8.3
Eeklo	A	6.5
Destelbergen	B	6.3
Landegem	B	5.2
Dendermonde	B	6.2
Eeklo	B	7.2

tidy data: principles

- Each **observation** forms a **row**
- Each **variable** forms a **column** and contains **values**

WWTP	Treatment	pH	Temperature (°C)
Destelbergen	A	8.	13.1
Landegem	A	7.5	16.9
Dendermonde	A	8.3	18.3
Eeklo	A	6.5	14.4
Destelbergen	B	6.3	17.2
Landegem	B	5.2	11.9
Dendermonde	B	6.2	17.1
Eeklo	B	7.2	19.0

tidy data: principles

- Each **observation** forms a **row**
- Each **variable** forms a **column** and contains **values**
- Each type of **observational unit** forms a **table**

WWTP	Treatment	pH	Temperature (°C)
Destelbergen	A	8.	13.1
Landegem	A	7.5	16.9
Dendermonde	A	8.3	18.3
Eeklo	A	6.5	14.4
...

WWTP	country	decimalLatitude	decimalLongitude
Dendermonde	BE	51.0248	4.136
Destelbergen	BE	51.051	3.774
Landegem	BE	51.052	3.568
...


Share your snippets during the coding session!

Go to <https://hackmd.io/QTJz1R1IRtyqdXamsGplfw> and post your code in between backticks:

For example:

```
```\n\nlibrary(lubridate)\n\nmy_data <- ... \n\n```
```

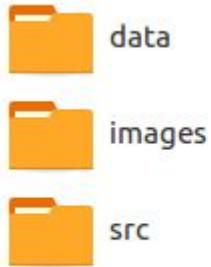
# The concept

We defined a number of challenges. If you were able to achieve a challenge, add a  to your laptop screen.

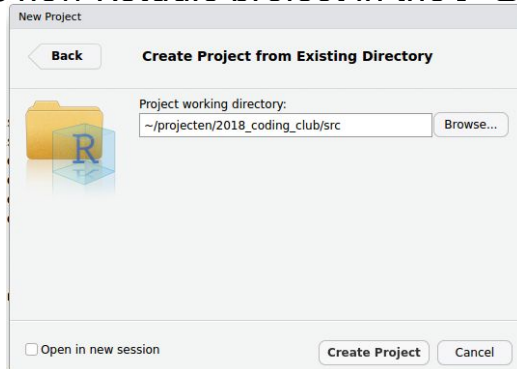
The objective is that **everyone** achieves !

- Someone has more  than you? **Ask for help!**
- Someone has less  than you? **Provide help!**

- Download coding club material from gdrive folder [INBO coding club](#) and **work locally, not in sync** with the Google drive



- Create new Rstudio project in the **/src** folder



- Download coding club material and work locally, not in sync with the Google drive
- Create new Rstudio project in the **src** folder...
- Use relative paths to data files!

Shared with me > INBO coding club > data 



Name ↓



 20190124\_tidy\_data\_representation\_collected\_data\_part2.xlsx 



 20190124\_tidy\_data\_representation\_collected\_data\_part1.xlsx 



 20181218\_bird\_rings.csv 

 20180821\_decay\_measurements\_3.csv 

 20180821\_decay\_measurements\_2.csv 

 20180821\_decay\_measurements\_1.csv 

 20180522\_gent\_groeiperwijk\_tidy.csv 

 20180426\_visdata\_cleaned.csv 

For this coding club:

[20190124\\_tidy\\_data\\_representation\\_collected\\_data\\_part1.xlsx](#)

[20190124\\_tidy\\_data\\_representation\\_collected\\_data\\_part2.xlsx](#)

[20190124\\_dryad\\_arias\\_hall\\_v3.csv](#)



Pair up with your neighbour.

Make the file [./data/20190124\\_survey\\_part1.xlsx](#) as tidy as possible:

1. Never modify the **raw** data: a (very) good practice
2. Document the issues you encountered in the [hackmd \(challenge 1\)](#)



Pair up with your neighbour.

Make the file [./data/20190124\\_survey\\_part2.xlsx](#) as tidy as possible:

1. Never modify the **raw** data: a (very) good practice
2. Document the issues you encountered in the [hackmd \(challenge 2\)](#)





Data that is easy to model,  
visualize and aggregate

Create the data you wish  
to see in the world



In this challenge we will make use of the open data, affiliated to the following [journal article](#):

Arias-Sánchez FI, Hall A (2016) Effects of antibiotic resistance alleles on bacterial evolutionary responses to viral parasites. *Biology Letters* 12(5): 20160064.  
<https://doi.org/10.1098/rsbl.2016.0064>



The experimental data of the main experiment of the paper, [20190124\\_dryad\\_arias\\_hall\\_v3.csv](#):

The screenshot shows the RStudio Source Editor with a data table. The table has 8 columns: AB\_r, Bacterial\_genotype, Phage\_t, OD\_0h, OD\_20h, OD\_72h, Survival\_72h, and PhageR\_72h. There are 6 rows of data.

	AB_r	Bacterial_genotype	Phage_t	OD_0h	OD_20h	OD_72h	Survival_72h	PhageR_72h
1	Rif	D516G	C_noPhage	0.1971	0.5960	0.6900	1	NA
2	Rif	D516G	C_noPhage	0.1593	0.5702	0.6989	1	NA
3	Rif	D516G	C_noPhage	0.0926	0.6613	0.6474	1	NA
4	Rif	D516G	C_noPhage	0.1482	0.6465	0.7045	1	NA
5	Rif	D516G	C_noPhage	0.0978	0.6752	0.6700	1	NA
6	Rif	D516G	C_noPhage	0.0897	0.6425	0.6846	1	NA

Actually, the columns OD\_0h, OD\_20h and OD\_72h are representing the same variable (i.e. optical\_density) and the column names itself represent a variable, i.e. experiment\_time\_h. We want to *tidy* these columns by converting them 2 columns: experiment\_time\_h and optical\_density.

Check the documentation on the [gather](#) function to do so. The script [20190124\\_challenge\\_3.R](#) will get you started.

**gather**(data, key, value, ..., na.rm = FALSE, convert = FALSE, factor\_key = FALSE)

gather() moves column names into a **key** column, gathering the column values into a single **value** column.

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

key value

```
gather(table4a, `1999`, `2000`,
key = "year", value = "cases")
```



the power of  
**GROUP\_BY**





Zaal: 01.70 - Ferdinand Peeters

Datum: 26-02-2019, van 10:00 tot 12:00

(registration announced via [DG\\_useR@inbo.be](mailto:DG_useR@inbo.be))