Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

Tidy Data

Hadley Wickham
RStudio

Hadley Wickham

Hadley Wickham in 2015

Lecture 5-2: Tidy data I

Dr. James Chen, Animal Data Scientist, School of Animal Sciences

Data Structure

> Happy families are all alike; every unhappy family is unhappy in its own way
>
> —Leo Tolstoy

Like families, tidy datasets are all alike but every messy dataset is messy in its own way. Tidy datasets provide a standardized way to link the structure of a dataset (its physical layout) with its semantics (its meaning). In this section, I'll provide some standard vocabulary for describing the structure and semantics of a dataset, and then use those definitions to define tidy data.

Data Semantics

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

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

Variable

A variable contains all values that measure the same underlying attribute (like height, temperature, duration) across units

Observation

An observation contains all values measured on the same unit (like a person, or a day, or a race) across attributes.

Data Semantics

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

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

Variable

A variable contains all values that measure the same underlying attribute (like height, temperature, duration) across units

Observation

An observation contains all values measured on the same unit (like a person, or a day, or a race) across attributes.

It is essential to understand what question you want to address from the data

Data Semantics

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

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

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

Variable

1. `person`, with three possible values (John, Mary, and Jane).
2. `treatment`, with two possible values (a and b).
3. `result`, with five or six values depending on how you think of the missing value (-, 16, 3, 2, 11, 1).

18 values: 3 variables, 6 observations

Data Semantics

Is it easy to figure out what are observations and what are variables?

	height	weight
Person 1	xxx	xxx
Person 2	xxx	xxx
Person 3	xxx	xxx

	height	width
Object 1	xxx	xxx
Object 2	xxx	xxx
Object 3	xxx	xxx

	dimension	value
Object 1	height	xxx
Object 1	width	xxx
Object 2	height	xxx
Object 2	width	xxx

Data Semantics

Is it easy to figure out what are observations and what are variables?

Variable

it is easier to describe functional relationships between **variables** (e.g., z is a linear combination of x and y, density is the ratio of weight to volume) than between rows

	height	width
Object 1	xxx	xxx
Object 2	xxx	xxx
Object 3	xxx	xxx

Observation

it is easier to make comparisons between groups of **observations** (e.g., average of group a vs. average of group b) than between groups of columns.

	dimension	value
Object 1	height	xxx
Object 1	width	xxx
Object 2	height	xxx
Object 2	width	xxx

Data Semantics

Is it easy to figure out what are observations and what are variables?

Variable

it is easier to describe functional relationships between **variables** (e.g., z is a linear combination of x and y, density is the ratio of weight to volume) than between rows

	Home phone	Work phone
Person 1	xxx	xxx
Person 2	xxx	xxx
Person 3	xxx	xxx

Observation

it is easier to make comparisons between groups of **observations** (e.g., average of group a vs. average of group b) than between groups of columns.

	Number type	Number
Person 1	work	xxx
Person 1	home	xxx
Person 2	work	xxx

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. → We can ignore this for now

Tidy data makes it easy for an analyst or a computer to extract needed variables because it provides a standard way of structuring a dataset. Compare Table 3 to Table 1: in Table 1 you need to use different strategies to extract different variables. This slows analysis and invites errors. If you consider how many data analysis operations involve all of the values in a variable (every aggregation function), you can see how important it is to extract these values in a simple, standard way. Tidy data is particularly well suited for vectorised programming languages like R, because the layout ensures that values of different variables from the same observation are always paired.

Tidying Messy Datasets

The five most common problems with messy datasets

Week 5 - 2

- Column headers are values, not variable names.
- Multiple variables are stored in one column.

Week 6 - 2

- Variables are stored in both rows and columns.
- Multiple types of observational units are stored in the same table.
- A single observational unit is stored in multiple tables.

Surprisingly, most messy datasets, including types of messiness not explicitly described above, can be tidied with a small set of tools: melting, string splitting, and casting. The following sections illustrate each problem with a real dataset that I have encountered, and show how to tidy them. The complete datasets and the R code used to tidy them are available online at `https://github.com/hadley/tidy-data`, and in the online supplementary materials for this paper.

Column Headers Are Values, Not Variable Names

United States® Census Bureau

2017 National population projection datasets

POP_5: population age 5 as of July 1

YEAR	POP_0	POP_1	POP_2	POP_3	POP_4
2016	3970145	3995008	3992154	3982074	3987656
2017	4054035	3982964	4008116	4003478	3992207
2018	4075563	4068172	3995888	4019345	4013649
2019	4095614	4089881	4082231	4006967	4029427
2020	4113164	4110117	4104058	4094281	4016919
2021	4127525	4127842	4124416	4116205	4105035
2022	4139039	4142382	4142254	4136660	4127020
2023	4147758	4154076	4156909	4154588	4147538
2024	4154108	4162971	4168717	4169332	4165525
2025	4158795	4169495	4177722	4181225	4180328
2026	4162506	4174374	4184383	4190355	4192326
2027	4165252	4178275	4189406	4197138	4201555
2028	4166643	4181210	4193447	4202280	4208440

What are the variables?

Year Age Population size

https://www.census.gov/data/datasets/2017/demo/popproj/2017-popproj.html

https://www2.census.gov/programs-surveys/popproj/technical-documentation/file-layouts/2017/np2017_d1.pdf

Column Headers Are Values, Not Variable Names

pandas.melt

```
pandas.melt(frame, id_vars=None, value_vars=None, var_name=None,
value_name='value', col_level=None, ignore_index=True)
```

Unpivot a DataFrame from wide to long format, optionally leaving identifiers set.

id_vars : *tuple, list, or ndarray, optional*

Column(s) to use as identifier variables.

value_vars : *tuple, list, or ndarray, optional*

Column(s) to unpivot. If not specified, uses all columns that are not set as *id_vars*.

var_name : *scalar* — age

Name to use for the 'variable' column. If None it uses `frame.columns.name` or 'variable'.

value_name : *scalar, default 'value'* — Pop. size

Name to use for the 'value' column.

id_vars — Identifier

value_vars — Columns to unpivot

YEAR	POP_0	POP_1	POP_2
2016	3970145	3995008	3992154
2017	4054035	3982964	4008116
2018	4075563	4068172	3995888
2019	4095614	4089881	4082231
2020	4113164	4110117	4104058
2021	4127525	4127842	4124416
2022	4139039	4142382	4142254
2023	4147758	4154076	4156909
2024	4154108	4162971	4168717
2025	4158795	4169495	4177722
2026	4162506	4174374	4184383
2027	4165252	4178275	4189406
2028	4166643	4181210	4193447

Imagine what its tidied form looks like …

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.melt.html

Column Headers Are Values, Not Variable Names


```
data_long = pd.melt(data,
                    id_vars=["YEAR"],
                    var_name="age",
                    value_name="pop")
data_long
```

✓ 0.0s

	YEAR	age	pop
0	2016	POP_0	3970145
1	2017	POP_0	4054035
2	2018	POP_0	4075563
3	2019	POP_0	4095614
4	2020	POP_0	4113164
...	...	...	...
4540	2056	POP_100	505951
4541	2057	POP_100	529280
4542	2058	POP_100	549748
4543	2059	POP_100	567379
4544	2060	POP_100	589382

[4545 rows x 3 columns]

id_vars
Identifier

value_vars
Columns to unpivot

YEAR	POP_0	POP_1	POP_2
2016	3970145	3995008	3992154
2017	4054035	3982964	4008116
2018	4075563	4068172	3995888
2019	4095614	4089881	4082231
2020	4113164	4110117	4104058
2021	4127525	4127842	4124416
2022	4139039	4142382	4142254
2023	4147758	4154076	4156909
2024	4154108	4162971	4168717
2025	4158795	4169495	4177722
2026	4162506	4174374	4184383
2027	4165252	4178275	4189406
2028	4166643	4181210	4193447

Imagine what its tidied form looks like …

Column Headers Are Values, Not Variable Names

Unpivot all columns

```
data_long = pd.melt(data,
                    id_vars=["YEAR"],
                    var_name="age",
                    value_name="pop")
data_long
```

✓ 0.0s

	YEAR	age	pop
0	2016	POP_0	3970145
1	2017	POP_0	4054035
2	2018	POP_0	4075563
3	2019	POP_0	4095614
4	2020	POP_0	4113164
...	...	...	...
4540	2056	POP_100	505951
4541	2057	POP_100	529280
4542	2058	POP_100	549748
4543	2059	POP_100	567379
4544	2060	POP_100	589382

[4545 rows x 3 columns]

Unpivot the first three columns

```
data_long = pd.melt(data,
                    id_vars=["YEAR"],
                    value_vars=["POP_1", "POP_2", "POP_3"],
                    var_name="age",
                    value_name="pop")
data_long
```

✓ 0.0s

	YEAR	age	pop
0	2016	POP_1	3995008
1	2017	POP_1	3982964
2	2018	POP_1	4068172
3	2019	POP_1	4089881
4	2020	POP_1	4110117
...	...	...	...
130	2056	POP_3	4401231
131	2057	POP_3	4411893
132	2058	POP_3	4421774
133	2059	POP_3	4430923
134	2060	POP_3	4439404

[135 rows x 3 columns]

Column Headers Are Values, Not Variable Names

Billboard top hits for 2000

id_vars
Identifier

value_vars
Columns to unpivot

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

What are the identifiers?

var_name
week

value_name
rank

week	rank
1	87
2	82
3	72

Column Headers Are Values, Not Variable Names

Billboard top hits for 2000

id_vars
Identifier

value_vars
Columns to unpivot

var_name
week

value_name
rank

Original

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

Tidied

year	artist	time	track	date	week	rank
2000	2 Pac	4:22	Baby Don't Cry	2000-02-26	1	87
2000	2 Pac	4:22	Baby Don't Cry	2000-03-04	2	82
2000	2 Pac	4:22	Baby Don't Cry	2000-03-11	3	72
2000	2 Pac	4:22	Baby Don't Cry	2000-03-18	4	77
2000	2 Pac	4:22	Baby Don't Cry	2000-03-25	5	87
2000	2 Pac	4:22	Baby Don't Cry	2000-04-01	6	94
2000	2 Pac	4:22	Baby Don't Cry	2000-04-08	7	99
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-02	1	91
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-09	2	87
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-16	3	92
2000	3 Doors Down	3:53	Kryptonite	2000-04-08	1	81
2000	3 Doors Down	3:53	Kryptonite	2000-04-15	2	70
2000	3 Doors Down	3:53	Kryptonite	2000-04-22	3	68
2000	3 Doors Down	3:53	Kryptonite	2000-04-29	4	67
2000	3 Doors Down	3:53	Kryptonite	2000-05-06	5	66

Column Headers Are Values, Not Variable Names

Billboard top hits for 2000

id_vars
Identifier

value_vars
Columns to unpivot

var_name
week

value_name
rank

Original

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

Tidied

year	artist	time	track	date	week	rank
2000	2 Pac	4:22	Baby Don't Cry	2000-02-26	1	87
2000	2 Pac	4:22	Baby Don't Cry	2000-03-04	2	82
2000	2 Pac	4:22	Baby Don't Cry	2000-03-11	3	72
2000	2 Pac	4:22	Baby Don't Cry	2000-03-18	4	77
2000	2 Pac	4:22	Baby Don't Cry	2000-03-25	5	87
2000	2 Pac	4:22	Baby Don't Cry	2000-04-01	6	94
2000	2 Pac	4:22	Baby Don't Cry	2000-04-08	7	99
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-02	1	91
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-09	2	87
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-16	3	92
2000	3 Doors Down	3:53	Kryptonite	2000-04-08	1	81
2000	3 Doors Down	3:53	Kryptonite	2000-04-15	2	70
2000	3 Doors Down	3:53	Kryptonite	2000-04-22	3	68
2000	3 Doors Down	3:53	Kryptonite	2000-04-29	4	67
2000	3 Doors Down	3:53	Kryptonite	2000-05-06	5	66

Column Headers Are Values, Not Variable Names

Billboard top hits for 2000

id_vars

Identifier

value_vars

Columns to unpivot

var_name

week

value_name

rank

Original

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

Tidied

year	artist	time	track	date	week	rank
2000	2 Pac	4:22	Baby Don't Cry	2000-02-26	1	87
2000	2 Pac	4:22	Baby Don't Cry	2000-03-04	2	82
2000	2 Pac	4:22	Baby Don't Cry	2000-03-11	3	72
2000	2 Pac	4:22	Baby Don't Cry	2000-03-18	4	77
2000	2 Pac	4:22	Baby Don't Cry	2000-03-25	5	87
2000	2 Pac	4:22	Baby Don't Cry	2000-04-01	6	94
2000	2 Pac	4:22	Baby Don't Cry	2000-04-08	7	99
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-02	1	91
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-09	2	87
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-16	3	92
2000	3 Doors Down	3:53	Kryptonite	2000-04-08	1	81
2000	3 Doors Down	3:53	Kryptonite	2000-04-15	2	70
2000	3 Doors Down	3:53	Kryptonite	2000-04-22	3	68
2000	3 Doors Down	3:53	Kryptonite	2000-04-29	4	67
2000	3 Doors Down	3:53	Kryptonite	2000-05-06	5	66

Column Headers Are Values, Not Variable Names

Billboard top hits for 2000

id_vars

Identifier

value_vars

Columns to unpivot

var_name

week

value_name

rank

Original

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

Tidied

year	artist	time	track	date	week	rank
2000	2 Pac	4:22	Baby Don't Cry	2000-02-26	1	87
2000	2 Pac	4:22	Baby Don't Cry	2000-03-04	2	82
2000	2 Pac	4:22	Baby Don't Cry	2000-03-11	3	72
2000	2 Pac	4:22	Baby Don't Cry	2000-03-18	4	77
2000	2 Pac	4:22	Baby Don't Cry	2000-03-25	5	87
2000	2 Pac	4:22	Baby Don't Cry	2000-04-01	6	94
2000	2 Pac	4:22	Baby Don't Cry	2000-04-08	7	99
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-02	1	91
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-09	2	87
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-16	3	92
2000	3 Doors Down	3:53	Kryptonite	2000-04-08	1	81
2000	3 Doors Down	3:53	Kryptonite	2000-04-15	2	70
2000	3 Doors Down	3:53	Kryptonite	2000-04-22	3	68
2000	3 Doors Down	3:53	Kryptonite	2000-04-29	4	67
2000	3 Doors Down	3:53	Kryptonite	2000-05-06	5	66

Multiple Variables Stored in One Column

Tuberculosis (TB) dataset

Gender + age range

country	year	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f014
AD	2000	0	0	1	0	0	0	0	—	—
AE	2000	2	4	4	6	5	12	10	—	3
AF	2000	52	228	183	149	129	94	80	—	93
AG	2000	0	0	0	0	0	0	1	—	1
AL	2000	2	19	21	14	24	19	16	—	3
AM	2000	2	152	130	131	63	26	21	—	1
AN	2000	0	0	1	2	0	0	0	—	0
AO	2000	186	999	1003	912	482	312	194	—	247
AR	2000	97	278	594	402	419	368	330	—	121
AS	2000	—	—	—	—	1	1	—	—	—

Multiple Variables Stored in One Column

Tuberculosis (TB) dataset

Original

id_vars value_vars

country	year	m014	m1524	m2534	m3544
AD	2000	0	0	1	0
AE	2000	2	4	4	6
AF	2000	52	228	183	149
AG	2000	0	0	0	0
AL	2000	2	19	21	14
AM	2000	2	152	130	131
AN	2000	0	0	1	2
AO	2000	186	999	1003	912
AR	2000	97	278	594	402
AS	2000	—	—	—	—

Tidied

id_vars value_vars

country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

var_name
column

value_name
cases

Multiple Variables Stored in One Column

Tuberculosis (TB) dataset

country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

country	year	sex	age	cases
AD	2000	m	0-14	0
AD	2000	m	15-24	0
AD	2000	m	25-34	1
AD	2000	m	35-44	0
AD	2000	m	45-54	0
AD	2000	m	55-64	0
AD	2000	m	65+	0
AE	2000	m	0-14	2
AE	2000	m	15-24	4
AE	2000	m	25-34	4
AE	2000	m	35-44	6
AE	2000	m	45-54	5
AE	2000	m	55-64	12
AE	2000	m	65+	10
AE	2000	f	0-14	3

Indexing — Get the first letter as the “gender” variable

Indexing — Get the last two numbers as the upper bound of the range