Tidy data;
Reshaping data to tidy it up

STAT-611

Tidy vs. Messy

Outline

1. What is tidy data?
2. How to make messy data tidy—the idea
3. How to make messy data tidy—examples
4. How to make messy data tidy—the tools

Based on Hadley Wickham's work. Search on

hadley wickham tidy data
hadley wickham reshape2

for more information

1. What is tidy data?

- The idea
  - We want data arranged in a way that is most useful for analysis
- Small data set example
  - N of days to recover from separate sore-throat events under each of two treatments:

name	treatmentA	treatmentB
John Smith	—	5
Jane Doe	1	4
Mary Johnson	2	3

Tidy data—definition

- For tidy data:
  1. Each variable forms a column,
  2. Each observation forms a row,
  3. Each table (or file) stores data about one class of experimental unit.
- If data is not tidy, it will be called messy
- This may still be vague
  1. What is a variable?
  2. What is “one class of experimental unit”?

More on variables

- It will also be useful to think of variables (columns) as being split into two groups:
  1. The *identifier (id) variables* identify the unit that measurements take place on.
     Usually discrete; often fixed by design. (In ANOVA notation (Yijk), id variables are the indices on the variables (i; j; k); in database notation, id variables are a composite primary key.)
  2. The *measured variables* represent what is measured on that unit (the Y's).

Id vs. Measured

- Which vars are id vars? Which are measured?

name	treatmentA	treatmentB
John Smith	—	5
Jane Doe	1	4
Mary Johnson	2	3

- Next: How many observations are on each row—1 or 2?

2. How to make messy data tidy—the idea

- If “experimental unit” here is meant to be associated with one subject and *at one point in time*, then each row contains two observations. (Assume the “if” is true here.)
- In that case the data is not tidy.
- How to make it tidy?

name	treatmentA	treatmentB
John Smith	—	5
Jane Doe	1	4
Mary Johnson	2	3

Tidy Data

- This data is now tidy according to the definition and assumptions:

name	treatment	value
John Smith	A	—
John Smith	B	5
Jane Doe	A	1
Jane Doe	B	4
Mary Johnson	A	2
Mary Johnson	B	3

- Missing data value? Keep or discard?
- Also, let's put id vars first, then measured vars

3. How to make messy data tidy: examples.

Is this data tidy or messy?

- First 10 rows of data on income/religion from a Pew Forum study. (5 last cols not shown ...)

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

Problem and Solution

- As in the first example: Column headers are values, not variable names. The fix (first 10 rows):

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

2 id vars, 1 measured var

- Often the multiple column headers may be time: week1, week2, …; Jan, Feb, …; 2001, 2002,…

Is this data tidy or messy?

- First 10 rows of data on tb (tuberculosis) counts for different countries, years, sexes, and ages

iso2	year	m014	m1524	...	m65	mu	f014
AD	2000	0	0		0		
AE	2000	2	4	...	10		3
AF	2000	52	228		80		93
AG	2000	0	0	...	1		1
AL	2000	2	19		16		3
AM	2000	2	152	...	21		1
AN	2000	0	0		0		0
AO	2000	186	999	...	194		247
AR	2000	97	278		330		121
AS	2000						

Problem and Solution

- As in the 2nd example: Some column headers are values, not variable names. In addition, the column headers represent more than one id var
- Solution →
  (4 id var's,
    1 measured var)

iso2	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

Is this data tidy or messy?

- First 10 rows and 5 days of data on min and max recorded temperatures at different locations and times (year, month, day)

id	year	month	element	d1	d2	d3	d4	d5
MX000017004	2010	1	tmax					
MX000017004	2010	1	tmin					
MX000017004	2010	2	tmax		273	241		
MX000017004	2010	2	tmin		144	144		
MX000017004	2010	3	tmax					321
MX000017004	2010	3	tmin					142
MX000017004	2010	4	tmax					
MX000017004	2010	4	tmin					
MX000017004	2010	5	tmax					
MX000017004	2010	5	tmin					

Problem and Solution

- Variables are stored in both rows and columns.
- Partial sol'n →
- But "element" contains *names of vars*, not values.

id	year	month	day	element	value
MX000017004	2010	1	30	tmax	278
MX000017004	2010	1	30	tmin	145
MX000017004	2010	2	2	tmax	273
MX000017004	2010	2	2	tmin	144
MX000017004	2010	2	3	tmax	241
MX000017004	2010	2	3	tmin	144
MX000017004	2010	2	11	tmax	297
MX000017004	2010	2	11	tmin	134
MX000017004	2010	2	23	tmax	299
MX000017004	2010	2	23	tmin	107

Problem and Solution

- Variables are stored in both rows and columns.
- Final sol'n →
- Four id vars, two measured vars.

id	year	month	day	tmax	tmin
MX000017004	2010	1	30	278	145
MX000017004	2010	2	2	273	144
MX000017004	2010	2	3	241	144
MX000017004	2010	2	11	297	134
MX000017004	2010	2	23	299	107
MX000017004	2010	3	5	321	142
MX000017004	2010	3	10	345	168
MX000017004	2010	3	16	311	176
MX000017004	2010	4	27	363	167
MX000017004	2010	5	27	332	182

4. How to make messy data tidy—the tools

- In all of these examples, we mostly have had to simply *reshape* the data
  - Many columns to one id var, from column names, and one measured var, from column values (income levels; days)
  - One column of var names to two column names (tmin, tmax)
  - Many columns to two id vars, from column names, (sex & age) and one measured var

Reshaping in R

- `reshape` function from the *stats* package
  - useful and standard, but we will not use in this course
- *reshape2* package, with primary functions
  - `melt`, to put a data frame into a canonical format
  - `dcast`, to reshape a melted data frame into desired form (`dcast`: o/p is data frame; `acast`:o/p is array)

  Also
  - `colsplit` helper function
  - use of `dcast` to summarize (aggregate) data in different shapes

What `melt` does

- Uses idea of id vars and measured vars:

	subject	time	age	weight	height
1	John Smith	1	33	90	2
2	Mary Smith	1			2

- Which vars are id? Which are measured?

What `melt` does

- Takes this one level of abstraction further, from this:

	subject	time	age	weight	height
1	John Smith	1	33	90	2
2	Mary Smith	1			2

- To this—"There are only id vars and a value var":

	subject	time	variable	value
1	John Smith	1	age	33
2	John Smith	1	weight	90
3	John Smith	1	height	2
4	Mary Smith	1	height	2

Sometimes called *molten data*

- This is *not* usually the final version of the data—instead, this is done to put things into a canonical (standard) form, for later processing by `dcast`.

What `dcast` does

- In these examples, all of the original id variables are used. (Note: *value* has a special meaning)

```
R> cast(smithsm, time + subject ~ variable)
  time    subject age weight height
1    1 John Smith  33     90   1.87
2    1 Mary Smith  NA     NA   1.54

R> cast(smithsm, ... ~ variable)
     subject time age weight height
1 John Smith    1  33     90   1.87
2 Mary Smith    1  NA     NA   1.54
```

- Left side of "~" → "down". Right side→"across"
- … → all other id vars

What `dcast` does

- More examples

```
R> cast(smithsm, ... ~ subject)
  time variable John Smith Mary Smith
1    1      age      33.00         NA
2    1   weight      90.00         NA
3    1   height       1.87       1.54

R> cast(smithsm, ... ~ time)
     subject variable     1
1 John Smith      age 33.00
2 John Smith   weight 90.00
3 John Smith   height  1.87
4 Mary Smith   height  1.54
```

- We will see more interesting examples in our R code

More information

- See tidy-data.pdf for
  - more examples
  - some other problems
  - some other ideas
- See reshape*.pdf for
  - some other ideas

- (Note: some function names have been updated since those papers were written)