HW3_tidy_data_instructions

Aidan O'Hara

2023-09-20

Instructions

Tidy the following tables from the Hadley Wickham, Tidy Data article into the given tidy (and melt) forms.

Only use the libraries included in the starter code:

```
library(foreign) #read.spss
library(stringr)
library(dplyr)
library(tidyr)
```

Use the included `dim(df) == c(x,y)` tests to check you have the right size data frame.

PREG

The starter code will the generate the raw data needed for the `preg` set. Use this as a good reminder that small portions of a given dataset can be used to first develop the tidy form before attempting to tidy the dataset at large.

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

Table 1: Typical presentation dataset.

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

Table 2: The same data as in Table 1 but with variables in columns and observations in rows.

PEW

This next table gets much more involved.

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

Table 3: The first ten rows of data on income and religion from the Pew Forum. Three columns, `$75-100k`, `$100-150k` and `>150k`, have been omitted

You will need to coerce some strings into the specified forms present in the tidy table.

Specifically, you will need to trim the `reltrad` column of `pew` from:

```
[1]  Evangelical Protestant Churches
[2]  Mainline Protestant Churches
[3]  Unaffiliated
[4]  Jewish
[5]  Don't know/refused (no information on religious affiliation)
[6]  Other Faiths
[7]  Historically Black Protestant Churches
[8]  Jehovah's Witness
[9]  Catholic
[10]  Buddhist
[11]  Mormon
[12]  Muslim
[13]  Hindu
[14]  Other Christian
[15]  Orthodox
[16]  Other World Religions
```

to:

```
[1] Evangelical Prot        Mainline Prot           Unaffiliated
[4] Jewish                  Don't know/refused      Other Faiths
[7] Historically Black Prot Jehovah's Witness       Atheist
[10] Agnostic                Catholic                Buddhist
[13] Mormon                  Muslim                  Hindu
[16] Other Christian         Orthodox                Other World Religions
```

and replace income values:

```
[1] 75 to under $100,000      20 to under $30,000       30 to under $40,000
[4] Less than $10,000         50 to under $75,000       $150,000 or more
[7] 40 to under $50,000       Don't know/Refused (VOL.) 100 to under $150,000
```

```
[10] 10 to under $20,000
```

with:

```
[1] "$75-100k"           "$20-30k"      "$30-40k"       "<$10k"
[5] "$50-75k"            ">150k"        "$40-50k"       "Don't know/refused"
[9] "$100-150k"          "$10-20k"
```

using the `stringr` library and dataframe column operations.

Once your strings are setup, factor the incomes, and use `count()` to create the tidy version.

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

Table 4: The first ten rows of the tidied Pew survey dataset on income and religion. The `column` has been renamed to `income`, and `value` to `freq`.

TB

For `tb`, starter code will take care of the first-step, minor, tidy-ups. Next, use `pivot_longer` to construct the molten version, before doing some additional clean-up/ rearrangement to reach the tidy format.

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	—	—	—

Table 5: Original TB dataset. Corresponding to each 'm' column for males, there is also an 'f' column for females, `f1524`, `f2534` and so on. These are not shown to conserve space. Note the mixture of 0s and missing values (—). This is due to the data collection process and the distinction is important for this 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

(a) Molten data

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

(b) Tidy data

Table 6: Tidying the TB dataset requires first melting, and then splitting the `column` column into two variables: `sex` and `age`.

WEATHER

The weather data, being particularly fickle, uses a custom `read` method provided in your starter code. Review the function and attempt to use the, base R, `read.fwf()` function to read in the raw data. What's the difference? Can `read.fwf()` be used instead?

id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
MX17004	2010	1	tmax	—	—	—	—	—	—	—	—
MX17004	2010	1	tmin	—	—	—	—	—	—	—	—
MX17004	2010	2	tmax	—	27.3	24.1	—	—	—	—	—
MX17004	2010	2	tmin	—	14.4	14.4	—	—	—	—	—
MX17004	2010	3	tmax	—	—	—	—	32.1	—	—	—
MX17004	2010	3	tmin	—	—	—	—	14.2	—	—	—
MX17004	2010	4	tmax	—	—	—	—	—	—	—	—
MX17004	2010	4	tmin	—	—	—	—	—	—	—	—
MX17004	2010	5	tmax	—	—	—	—	—	—	—	—
MX17004	2010	5	tmin	—	—	—	—	—	—	—	—

Table 7: Original weather dataset. There is a column for each possible day in the month. Columns `d9` to `d31` have been omitted to conserve space.

After reading in the raw data you'll need to do the following to create the above table and melt the data:

- Subset for just the year 2010, and only elements `"TMAX"` and `"TMIN"`
- Remove all columns besides `id`, `year`, `month`, `element`, and columns beginning with `value`
- Remove the extra `0`s from the `id` column
- Change the column names beginning with `value`, ex `value_1`, to `d`, example `d1`
- Divide the values in the new `d` columns by 10
- Finally, make values of the `element` column lowercase

A quick `pivot_longer()` and voila, you've made `melt_weather`.

Translate the new `variable` column from `dn` to just `n`, and convert the year, month, and day data into a single column of dates. Use `as.date()` and `ISOdate()` accordingly.

Don't forget to select and arrange your columns to match the table below.

One more thing to do now, use a tidyr pivot to construct the final `tidy_weather` dataframe!

id	date	element	value
MX17004	2010-01-30	tmax	27.8
MX17004	2010-01-30	tmin	14.5
MX17004	2010-02-02	tmax	27.3
MX17004	2010-02-02	tmin	14.4
MX17004	2010-02-03	tmax	24.1
MX17004	2010-02-03	tmin	14.4
MX17004	2010-02-11	tmax	29.7
MX17004	2010-02-11	tmin	13.4
MX17004	2010-02-23	tmax	29.9
MX17004	2010-02-23	tmin	10.7

(a) Molten data

id	date	tmax	tmin
MX17004	2010-01-30	27.8	14.5
MX17004	2010-02-02	27.3	14.4
MX17004	2010-02-03	24.1	14.4
MX17004	2010-02-11	29.7	13.4
MX17004	2010-02-23	29.9	10.7
MX17004	2010-03-05	32.1	14.2
MX17004	2010-03-10	34.5	16.8
MX17004	2010-03-16	31.1	17.6
MX17004	2010-04-27	36.3	16.7
MX17004	2010-05-27	33.2	18.2

(b) Tidy data

Table 8: (a) Molten weather dataset. This is almost tidy, but instead of values, the `element` column contains names of variables. Missing values are dropped to conserve space. (b) Tidy weather dataset. Each row represents the meteorological measurements for a single day. There are two measured variables, minimum (`tmin`) and maximum (`tmax`) temperature; all other variables are fixed.