Tidy data

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair

Department of Statistics / Rice University

June 2012

1. What is tidy data?

2. Five common causes of messiness

3. Tidying messy data (x5)

What is tidy data?

- A step along the road to clean data
- Data that is easy to model, visualise and aggregate (i.e. works well with `lm`, `ggplot`, and `ddply`)
- Variables in columns, observations in rows, one type per dataset

	Pregnant	Not pregnant
Male	0	5
Female	1	4

There are three variables in this data set. What are they?

pregnant	sex	n
no	female	4
no	male	5
yes	female	1
yes	male	0

Storage	Meaning
Table / File	Data set
Rows	Observations
Columns	Variables

Causes of messiness

- Column headers are values, not variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of experimental unit stored in the same table
- One type of experimental unit stored in multiple tables

```
# Tools

library(reshape2)
?melt
?dcast
?col_split

library(stringr)
?str_replace
?str_sub
?str_split_fixed

library(plyr)
?arrange
```

Column headers values, not variable names

Income distribution within US religious groups

- Survey data that examines the relationship between income and religious affiliation
- collected by the Pew Forum on Religious and Public life http://pewforum.org/Income-Distribution-Within-US-Religious-Groups.aspx

	religion	<$10k	$10-20k	$20-30k	$30-40k	$40-50k	$50-75k
1	Agnostic	27	34	60	81	76	137
2	Atheist	12	27	37	52	35	70
3	Buddhist	27	21	30	34	33	58
4	Catholic	418	617	732	670	638	1116
5	Don’t know/refused	15	14	15	11	10	35
6	Evangelical Prot	575	869	1064	982	881	1486
7	Hindu	1	9	7	9	11	34
8	Historically Black Prot	228	244	236	238	197	223
9	Jehovah’s Witness	20	27	24	24	21	30
10	Jewish	19	19	25	25	30	95
11	Mainline Prot	289	495	619	655	651	1107
12	Mormon	29	40	48	51	56	112
13	Muslim	6	7	9	10	9	23
14	Orthodox	13	17	23	32	32	47
15	Other Christian	9	7	11	13	13	14
16	Other Faiths	20	33	40	46	49	63
17	Other World Religions	5	2	3	4	2	7
18	Unaffiliated	217	299	374	365	341	528

	religion	<$10k	$10-20k	$20-30k	$30-40k	$40-50k	$50-75k
1	Agnostic	27	34	60	81	76	137
2	Atheist	12	27	37	52	35	70
3	Buddhist	27	21	30	34	33	58
4	Catholic	418	617	732	670	638	1116
5	Don’t know/refused	15	14	15	11	10	35
6	Evangelical Prot	575	869	1064	982	881	1486
7	Hindu	1	9	7	9	11	34
8	Historically Black Prot	228	244	236	238	197	223
9	Jehovah’s Witness	20	27	24	24	21	30
10	Jewish	19	19	25	25	30	95
11	Mainline Prot	289	495	619	655	651	1107
12	Mormon	29	40	48	51	56	112
13	Muslim	6	7	9	10	9	23
14	Orthodox	13	17	23	32	32	47
15	Other Christian	9	7	11	13	13	14
16	Other Faiths	20	33	40	46	49	63
17	Other World Religions	5	2	3	4	2	7
18	Una						

```
# What are the variables in this dataset?
# Discuss with your neighbour for 1 minute
```

```
raw <- read.delim("pew.txt", check.names = F,
 stringsAsFactors = F)
```

```
head(raw)
```

```
# Fixing this problem is easy.  We use melt, from
# reshape2, with two arguments, the input data, and
# the columns which are already variables:

library(reshape2)
tidy <- melt(raw, "religion")
```

```
head(tidy)

# We can now tweak the variable names
names(tidy) <- c("religion", "income", "n")
```

Multiple variables in one column

	iso2	year	m04	m514	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f04	f514	f014
1	AD	1989	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	AD	1990	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	AD	1991	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	AD	1992	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
5	AD	1993	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
6	AD	1994	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
7	AD	1996	NA	NA	0	0	0	4	1	0	0	NA	NA	NA	0
8	AD	1997	NA	NA	0	0	1	2	2	1	6	NA	NA	NA	0
9	AD	1998	NA	NA	0	0	0	1	0	0	0	NA	NA	NA	NA
10	AD	1999	NA	NA	0	0	0	1	1	0	0	NA	NA	NA	0
11	AD	2000	NA	NA	0	0	1	0	0	0	0	NA	NA	NA	NA
12	AD	2001	NA	NA	0	NA	NA	2	1	NA	NA	NA	NA	NA	NA
13	AD	2002	NA	NA	0	0	0	1	0	0	0	NA	NA	NA	0
14	AD	2003	NA	NA	0	0	0	1	2	0	0	NA	NA	NA	0
15	AD	2004	NA	NA	0	0	0	1	1	0	0	NA	NA	NA	0
16	AD	2005	0	0	0	0	1	1	0	0	0	0	0	0	0
17	AD	2006	0	0	0	1	1	2	0	1	1	0	0	0	0
18	AD	2007	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
19	AD	2008	0	0	0	0	0	0	1	0	0	0	0	0	0
20	AE	1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

	iso2	year	m04	m514	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f04	f514	f014
1	AD	1989	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	AD	1990	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	AD	1991	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	AD	1992	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
5	AD	1993	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
6	AD	1994	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
7	AD	1996	NA	NA	0	0	0	4	1	0	0	NA	NA	NA	0
8	AD	1997	NA	NA	0	0	1	2	2	1	6	NA	NA	NA	0
9	AD	1998	NA	NA	0	0	0	1	0	0	0	NA	NA	NA	NA
10	AD	1999	NA	NA	0	0	0	1	1	0	0	NA	NA	NA	0
11	AD	2000	NA	NA	0	0	1	0	0	0	0	NA	NA	NA	NA
12	AD	2001	NA	NA	0	NA	NA	2	1	NA	NA	NA	NA	NA	NA
13	AD	2002	NA	NA	0	0	0	1	0	0	0	NA	NA	NA	0
14	AD	2003	NA	NA	0	0	0	1	2	0	0	NA	NA	NA	0
15	AD	2004	NA	NA	0	0	0	1	1	0	0	NA	NA	NA	0
16	AD	2005	0	0	0	0	1	1	0	0	0	0	0	0	0
17	AD	2006	0	0	0	1	1	2	0	1	1	0	0	0	0
18	AD	2007	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
19	AD	2008													
20	AE	1980													

```
# What are the variables in this dataset?
# Discuss with your neighbour for 1 minute
# Hint: f = female, u = unknown, 1524 = 15-24
```

```
raw <- read.csv("tb.csv", stringsAsFactors = FALSE)
raw$new_sp <- NULL
```

```
names(raw) <- str_replace(names(raw), "new_sp_", "")
```

Your turn

Use melt in the same way as for the religion-income data to get all variables in columns.

Think about how you might separate the "variable" variable into age and sex.

```
# na.rm = TRUE is useful if the missings don't have
# any meaning
tidy <- melt(raw, id = c("iso2", "year"),
  na.rm = TRUE)
names(tidy)[4] <- "cases"
```

```
# Often a good idea to ensure the rows are ordered
# by the variables
tidy <- arrange(tidy, iso2, variable, year)
```

```
str_sub(tidy$variable, 1, 1)
str_sub(tidy$variable, 2)


ages <- c("04" = "0-4", "514" = "5-14", "014" =
"0-14", "1524" = "15-24", "2534" = "25-34", "3544" =
"35-44", "4554" = "45-54", "5564" = "55-64", "65"=
"65+", "u" = NA)
ages[str_sub(tidy$variable, 2)]


tidy$sex <- str_sub(tidy$variable, 1, 1)
tidy$age <- factor(ages[str_sub(tidy$variable, 2)],
   levels = ages)
tidy$variable <- NULL

tidy <- tidy[c("iso2", "year", "sex", "age", "cases")]
```

Variables in rows and columns

	id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12
1	MX000017004	2010	1	TMAX	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	MX000017004	2010	1	TMIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	MX000017004	2010	2	TMAX	NA	273	241	NA	NA	NA	NA	NA	NA	NA	297	NA
4	MX000017004	2010	2	TMIN	NA	144	144	NA	NA	NA	NA	NA	NA	NA	134	NA
5	MX000017004	2010	3	TMAX	NA	NA	NA	NA	321	NA	NA	NA	NA	345	NA	NA
6	MX000017004	2010	3	TMIN	NA	NA	NA	NA	142	NA	NA	NA	NA	168	NA	NA
7	MX000017004	2010	4	TMAX	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
8	MX000017004	2010	4	TMIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
9	MX000017004	2010	5	TMAX	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
10	MX000017004	2010	5	TMIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	MX000017004	2010	6	TMAX	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
12	MX000017004	2010	6	TMIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
13	MX000017004	2010	7	TMAX	NA	NA	286	NA	NA	NA	NA	NA	NA	NA	NA	NA
14	MX000017004	2010	7	TMIN	NA	NA	175	NA	NA	NA	NA	NA	NA	NA	NA	NA
15	MX000017004	2010	8	TMAX	NA	NA	NA	NA	296	NA	NA	290	NA	NA	NA	NA
16	MX000017004	2010	8	TMIN	NA	NA	NA	NA	158	NA	NA	173	NA	NA	NA	NA
17	MX000017004	2010	10	TMAX	NA	NA	NA	NA	270	NA	281	NA	NA	NA	NA	NA
18	MX000017004	2010	10	TMIN	NA	NA	NA	NA	140	NA	129	NA	NA	NA	NA	NA
19	MX000017004	2010	11	TMAX	NA	313	NA	272	263	NA	NA	NA	NA	NA	NA	NA
20	MX000017004	2010	11	TMIN	NA	163	NA	120	79	NA	NA	NA	NA	NA	NA	NA

	id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12
1	MX000017004	2010	1	TMAX	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	MX000017004	2010	1	TMIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	MX000017004	2010	2	TMAX	NA	273	241	NA	NA	NA	NA	NA	NA	NA	297	NA
4	MX000017004	2010	2	TMIN	NA	144	144	NA	NA	NA	NA	NA	NA	NA	134	NA
5	MX000017004	2010	3	TMAX	NA	NA	NA	NA	321	NA	NA	NA	NA	345	NA	NA
6	MX000017004	2010	3	TMIN	NA	NA	NA	NA	142	NA	NA	NA	NA	168	NA	NA
7	MX000017004	2010	4	TMAX	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
8	MX000017004	2010	4	TMIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
9	MX000017004	2010	5	TMAX	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
10	MX000017004	2010	5	TMIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	MX000017004	2010	6	TMAX	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
12	MX000017004	2010	6	TMIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
13	MX000017004	2010	7	TMAX	NA	NA	286	NA	NA	NA	NA	NA	NA	NA	NA	NA
14	MX000017004	2010	7	TMIN	NA	NA	175	NA	NA	NA	NA	NA	NA	NA	NA	NA
15	MX000017004	2010	8	TMAX	NA	NA	NA	NA	296	NA	NA	290	NA	NA	NA	NA
16	MX000017004	2010	8	TMIN	NA	NA	NA	NA	158	NA	NA	173	NA	NA	NA	NA
17	MX000017004	201														
18	MX000017004	201														
19	MX000017004	201														
20	MX000017004	201														

```
# What are the variables in this dataset?
# Discuss with your neighbour for 1 minute
# Hint: TMIN = minimum temperature,
#       id = weather station identifier
```

```
raw <- read.delim("weather.txt",
    stringsAsFactors = FALSE)
```

Your turn

Melt the data, clean variables, and reorder rows and columns.

What do you need to do next?

```
raw1 <- melt(raw, id = 1:4, na.rm = T)
raw1$day <- as.integer(
  str_replace(raw1$variable, "d", ""))
raw1$variable <- NULL
raw1$element <- tolower(raw1$element)
```

```
raw1 <- raw1[c("id", "year", "month", "day",
  "element", "value")]
raw1 <- arrange(raw1, year, month, day, element)
```

```
# dcast shifts variables from rows to columns
tidy <- dcast(raw1, ... ~ element)

# casting syntax:
#   row_var1 + row_var2 ~ col_var1 + col_var2
#   ... = all variables not otherwise mentioned
```

Multiple types in the same table

Your turn

Practice everything you've learned so far to tidy up `billboard.csv`.

(You might want to peek in `billboard-encoding.r`)

```
raw <- read.csv("billboard.csv",
  stringsAsFactors = F)
raw$date.peaked <- NULL
raw$artist.inverted <- iconv(raw$artist.inverted,
  "MAC", "UTF-8")
raw$track <- str_replace(raw$track,
  " \\(.*?\\)", "")
names(raw)[-(1:6)] <- 1:76

tidy <- melt(raw, 1:6, na.rm = T)
tidy$week <- as.integer(as.character(tidy$variable))
tidy$variable <- NULL
```

```
# Fix dates (bonus)
library(lubridate)
tidy$date.entered <- ymd(tidy$date.entered)
tidy$date <- tidy$date.entered +
  weeks(tidy$week - 1)
tidy$date.entered <- NULL

# Tidy column names, order and row order
tidy <- rename(tidy, c("value" = "rank",
  "artist.inverted" = "artist"))
tidy <- tidy[c("year", "artist", "track", "time",
   "genre", "week", "date", "rank")]
tidy <- arrange(tidy, year, artist, track, week)



tidy <- tidy[c("year", "date", "artist", "track", "time",
   "genre", "week", "rank")]
tidy <- arrange(tidy, year, date, artist, track)
```

Normalisation

Each fact about a song is repeated many many times. Sign that multiple types of experimental unit stored in the same table. We can store our data more efficiently by separating it into different tables for each type of unit.

Need to separate out into song and rank tables.

```
song <- unrowname(unique(tidy[c("artist", "track",
"genre", "time")]))
song$song_id <- 1:nrow(song)
```

```
rank <- join(tidy, song, match = "first")
rank <- rank[c("song_id", "date", "rank")]
```

One type in multiple tables

```
# Not shown, but easy with ldply
files <- dir("path", pattern = ".csv", full = T)
names(files) <- basename(files)

all <- ldply(files, read.csv)
```